

Appendix F

/* IR after various phases for first part of program*/

IR after CIL Reader (flag cil)

{-7}	= START _try	#37
_i, _q, _a, _b	= ENTERFUNC _try	#37
_j	= ASSIGN 0	#40
t110	= ASSIGN _q	#41
[t110]	= ASSIGN 0	#41
	GOTO \$L2	#42
\$L2: (2 refs)		#42
t111	= ASSIGN _q	#42
t112	= CMP(EQ) [t111], 0	#42
	CBRANCH(EQ) t112, \$L4, \$L3	#42
\$L4: (1 ref)		#42
t113	= CMP(NE) _j, 8	#42
	CBRANCH(NE) t113, \$L5, \$L3	#42
\$L5: (1 ref)		#42
t114	= ADD _j, 1	#44
_j	= ASSIGN t114	#44
t115	= ASSIGN _q	#45
[t115]	= ASSIGN 0	#45
t116	= MUL _j, 4	#46
t117	= ADD _b, t116	#46
t118	= ASSIGN t117	#46
t119	= CMP(EQ) [t118], 1	#46
	CBRANCH(EQ) t119, \$L7, \$L6	#46
\$L7: (1 ref)		#46
t120	= ADD _i, _j	#46
t121	= MUL t120, 4	#46
t122	= ADD _a, t121	#46
t123	= ASSIGN t122	#46
t124	= CMP(EQ) [t123], 1	#46
	CBRANCH(EQ) t124, \$L8, \$L6	#46
\$L8: (1 ref)		#46
t126	= SUB _i, _j	#46
t127	= ADD t126, 7	#46
t128	= MUL t127, 4	#46
t129	= SUBSCRIPT &_c, t128	#46
t130	= ASSIGN t129	#46
t131	= CMP(EQ) [t130], 1	#46
	CBRANCH(EQ) t131, \$L9, \$L6	#46
\$L9: (1 ref)		#46
t133	= MUL _i, 4	#48
t134	= SUBSCRIPT &_x, t133	#48
t135	= ASSIGN t134	#48
[t135]	= ASSIGN _j	#48
t136	= MUL _j, 4	#49
t137	= ADD _b, t136	#49
t138	= ASSIGN t137	#49
[t138]	= ASSIGN 0	#49
t139	= ADD _i, _j	#50

AppendixF.txt

t140	= MUL t139, 4	#50
t141	= ADD _a, t140	#50
t142	= ASSIGN t141	#50
[t142]	= ASSIGN 0	#50
t143	= SUB _i, _j	#51
t144	= ADD t143, 7	#51
t145	= MUL t144, 4	#51
t146	= SUBSCRIPT &_c, t145	#51
t147	= ASSIGN t146	#51
[t147]	= ASSIGN 0	#51
t148	= CMP(LT) _i, 8	#52
	CBRANCH(LT) t148, \$L11, \$L10	#52
\$L11: (1 ref)		#52
t149	= ADD _i, 1	#54
{-5}	= CALL _try, t149, _q, _a, _b, {-5} [Handler: \$L1]	#54
t151	= ASSIGN _q	#55
t152	= CMP(EQ) [t151], 0	#55
	CBRANCH(EQ) t152, \$L13, \$L12	#55
\$L13: (1 ref)		#55
t153	= MUL _j, 4	#57
t154	= ADD _b, t153	#57
t155	= ASSIGN t154	#57
[t155]	= ASSIGN 1	#57
t156	= ADD _i, _j	#58
t157	= MUL t156, 4	#58
t158	= ADD _a, t157	#58
t159	= ASSIGN t158	#58
[t159]	= ASSIGN 1	#58
t160	= SUB _i, _j	#59
t161	= ADD t160, 7	#59
t162	= MUL t161, 4	#59
t163	= SUBSCRIPT &_c, t162	#59
t164	= ASSIGN t163	#59
[t164]	= ASSIGN 1	#59
	GOTO \$L12	#55
\$L12: (2 refs)		#55
	GOTO \$L14	#62
\$L10: (1 ref)		#52
t165	= ASSIGN _q	#62
[t165]	= ASSIGN 1	#62
	GOTO \$L14	#62
\$L14: (2 refs)		#62
	GOTO \$L6	#46
\$L6: (4 refs)		#46
	GOTO \$L2	#64
\$L3: (2 refs)		#42
	GOTO \$L15	#65
\$L1: (1 ref)		#37
	UNWIND _try	#37
\$L15: (1 ref)		#65
	EXITFUNC _try	#65
	END _try, {-7}	#65

AppendixF.txt

IR after Type Checker (flag Type Checker)

```

    {-7}          = START _try                                #37
    _i, _q, _a, _b = ENTERFUNC _try                          #37
    _j            = ASSIGN 0                                  #40
    t110          = ASSIGN _q                                 #41
    [t110]        = ASSIGN 0                                  #41
                                GOTO $L2                      #42
$L2: (2 refs)                                           #42
    t111          = ASSIGN _q                                 #42
    t112          = CMP(EQ) [t111], 0                        #42
                                CBRANCH(EQ) t112, $L4, $L3    #42
$L4: (1 ref)                                           #42
    t113          = CMP(NE) _j, 8                            #42
                                CBRANCH(NE) t113, $L5, $L3    #42
$L5: (1 ref)                                           #42
    t114          = ADD _j, 1                                 #44
    _j            = ASSIGN t114                              #44
    t115          = ASSIGN _q                                 #45
    [t115]        = ASSIGN 0                                  #45
    t116          = MUL _j, 4                                 #46
    t117          = ADD _b, t116                             #46
    t118          = ASSIGN t117                              #46
    t119          = CMP(EQ) [t118], 1                        #46
                                CBRANCH(EQ) t119, $L7, $L6    #46
$L7: (1 ref)                                           #46
    t120          = ADD _i, _j                                #46
    t121          = MUL t120, 4                               #46
    t122          = ADD _a, t121                             #46
    t123          = ASSIGN t122                              #46
    t124          = CMP(EQ) [t123], 1                        #46
                                CBRANCH(EQ) t124, $L8, $L6    #46
$L8: (1 ref)                                           #46
    t126          = SUB _i, _j                                #46
    t127          = ADD t126, 7                               #46
    t128          = MUL t127, 4                               #46
    t129          = SUBSCRIPT &_c, t128                     #46
    t130          = ASSIGN t129                              #46
    t131          = CMP(EQ) [t130], 1                        #46
                                CBRANCH(EQ) t131, $L9, $L6    #46
$L9: (1 ref)                                           #46
    t133          = MUL _i, 4                                 #48
    t134          = SUBSCRIPT &_x, t133                      #48
    t135          = ASSIGN t134                              #48
    [t135]        = ASSIGN _j                                #48
    t136          = MUL _j, 4                                 #49
    t137          = ADD _b, t136                             #49
    t138          = ASSIGN t137                              #49
    [t138]        = ASSIGN 0                                  #49
    t139          = ADD _i, _j                                #50
    t140          = MUL t139, 4                               #50
    t141          = ADD _a, t140                             #50

```

AppendixF.txt

t142	= ASSIGN t141	#50
[t142]	= ASSIGN 0	#50
t143	= SUB _i, _j	#51
t144	= ADD t143, 7	#51
t145	= MUL t144, 4	#51
t146	= SUBSCRIPT &_c, t145	#51
t147	= ASSIGN t146	#51
[t147]	= ASSIGN 0	#51
t148	= CMP(LT) _i, 8	#52
	CBRANCH(LT) t148, \$L11, \$L10	#52
\$L11: (1 ref)		#52
t149	= ADD _i, 1	#54
{-5}	= CALL _try, t149, _q, _a, _b, {-5} [Handler: \$L1]	#54
t151	= ASSIGN _q	#55
t152	= CMP(EQ) [t151], 0	#55
	CBRANCH(EQ) t152, \$L13, \$L12	#55
\$L13: (1 ref)		#55
t153	= MUL _j, 4	#57
t154	= ADD _b, t153	#57
t155	= ASSIGN t154	#57
[t155]	= ASSIGN 1	#57
t156	= ADD _i, _j	#58
t157	= MUL t156, 4	#58
t158	= ADD _a, t157	#58
t159	= ASSIGN t158	#58
[t159]	= ASSIGN 1	#58
t160	= SUB _i, _j	#59
t161	= ADD t160, 7	#59
t162	= MUL t161, 4	#59
t163	= SUBSCRIPT &_c, t162	#59
t164	= ASSIGN t163	#59
[t164]	= ASSIGN 1	#59
	GOTO \$L12	#55
\$L12: (2 refs)		#55
	GOTO \$L14	#62
\$L10: (1 ref)		#52
t165	= ASSIGN _q	#62
[t165]	= ASSIGN 1	#62
	GOTO \$L14	#62
\$L14: (2 refs)		#62
	GOTO \$L6	#46
\$L6: (4 refs)		#46
	GOTO \$L2	#64
\$L3: (2 refs)		#42
	GOTO \$L15	#65
\$L1: (1 ref)		#37
	UNWIND _try	#37
\$L15: (1 ref)		#65
	EXITFUNC _try	#65
	END _try, {-7}	#65

AppendixF.txt

```

{-7}          = START _try                                #37
_i, _q, _a, _b = ENTERFUNC _try                            #37
_j            = ASSIGN 0                                    #40
t110          = ASSIGN _q                                    #41
[t110]        = ASSIGN 0                                    #41
              GOTO $L2                                       #42
$L2: (2 refs)                                           #42
t111          = ASSIGN _q                                    #42
t112          = CMP(EQ) [t111], 0                          #42
              CBRANCH(EQ) t112, $L4, $L3                    #42
$L4: (1 ref)                                           #42
t113          = CMP(NE) _j, 8                               #42
              CBRANCH(NE) t113, $L5, $L3                    #42
$L5: (1 ref)                                           #42
t114          = ADD _j, 1                                    #44
_j            = ASSIGN t114                                  #44
t115          = ASSIGN _q                                    #45
[t115]        = ASSIGN 0                                    #45
t116          = MUL _j, 4                                    #46
t117          = ADD _b, t116                                 #46
t118          = ASSIGN t117                                  #46
t119          = CMP(EQ) [t118], 1                           #46
              CBRANCH(EQ) t119, $L7, $L6                    #46
$L7: (1 ref)                                           #46
t120          = ADD _i, _j                                    #46
t121          = MUL t120, 4                                  #46
t122          = ADD _a, t121                                 #46
t123          = ASSIGN t122                                  #46
t124          = CMP(EQ) [t123], 1                           #46
              CBRANCH(EQ) t124, $L8, $L6                    #46
$L8: (1 ref)                                           #46
t126          = SUB _i, _j                                    #46
t127          = ADD t126, 7                                  #46
t128          = MUL t127, 4                                  #46
t129          = ADD &_c, t128                                #46
t130          = ASSIGN t129                                  #46
t131          = CMP(EQ) [t130], 1                           #46
              CBRANCH(EQ) t131, $L9, $L6                    #46
$L9: (1 ref)                                           #46
t133          = MUL _i, 4                                    #48
t134          = ADD &_x, t133                                #48
t135          = ASSIGN t134                                  #48
[t135]        = ASSIGN _j                                    #48
t136          = MUL _j, 4                                    #49
t137          = ADD _b, t136                                 #49
t138          = ASSIGN t137                                  #49
[t138]        = ASSIGN 0                                    #49
t139          = ADD _i, _j                                    #50
t140          = MUL t139, 4                                  #50
t141          = ADD _a, t140                                 #50
t142          = ASSIGN t141                                  #50
[t142]        = ASSIGN 0                                    #50

```

AppendixF.txt

t143	= SUB _i, _j	#51
t144	= ADD t143, 7	#51
t145	= MUL t144, 4	#51
t146	= ADD &_c, t145	#51
t147	= ASSIGN t146	#51
[t147]	= ASSIGN 0	#51
t148	= CMP(LT) _i, 8	#52
	CBRANCH(LT) t148, \$L11, \$L10	#52
\$L11: (1 ref)		#52
t149	= ADD _i, 1	#54
{-5}	= CALL _try, t149, _q, _a, _b, {-5} [Handler: \$L1]	#54
t151	= ASSIGN _q	#55
t152	= CMP(EQ) [t151], 0	#55
	CBRANCH(EQ) t152, \$L13, \$L12	#55
\$L13: (1 ref)		#55
t153	= MUL _j, 4	#57
t154	= ADD _b, t153	#57
t155	= ASSIGN t154	#57
[t155]	= ASSIGN 1	#57
t156	= ADD _i, _j	#58
t157	= MUL t156, 4	#58
t158	= ADD _a, t157	#58
t159	= ASSIGN t158	#58
[t159]	= ASSIGN 1	#58
t160	= SUB _i, _j	#59
t161	= ADD t160, 7	#59
t162	= MUL t161, 4	#59
t163	= ADD &_c, t162	#59
t164	= ASSIGN t163	#59
[t164]	= ASSIGN 1	#59
	GOTO \$L12	#55
\$L12: (2 refs)		#55
	GOTO \$L14	#62
\$L10: (1 ref)		#52
t165	= ASSIGN _q	#62
[t165]	= ASSIGN 1	#62
	GOTO \$L14	#62
\$L14: (2 refs)		#62
	GOTO \$L6	#46
\$L6: (4 refs)		#46
	GOTO \$L2	#64
\$L3: (2 refs)		#42
	GOTO \$L15	#65
\$L1: (1 ref)		#37
	UNWIND _try	#37
\$L15: (1 ref)		#65
	EXITFUNC _try	#65
	END _try, {-7}	#65

IR after Ssa Construction and Optimization (flag Ssa)

==== Block 1 Pred() Succ(2) next 2 pre 1 post 38 iDom 1 df

AppendixF.txt

```

{-7}, {-1} = START _try #37
==== Block 2 Pred(1) Succ(3) prev 1 next 3 pre 2 post 37 iDom 1 df
_i<*4>, _q<*3>, _a<*2>, _b<*1> = ENTERFUNC _try #37
_j<*5> = ASSIGN 0 #40
tv110-<*8> = ASSIGN _q<3> #41
[tv110-<8>] = ASSIGN 0 #41
GOTO $L2 #42
==== Block 3 Pred(15,2) Succ(16,4) prev 2 next 4 pre 3 post 36 iDom 2 df 3
$L2: (2 refs) #42
_j<*6> = PHI _j<7>, _j<5> #65
tv111-<*9> = ASSIGN _q<3> #42
tv112-<*10> = CMP(EQ) [tv111-<9>], 0 #42
CBRANCH(EQ) tv112-<10>, $L4, $L3 #42
==== Block 4 Pred(3) Succ(16,5) prev 3 next 5 pre 10 post 35 iDom 3 df 3,16,
19
$L4: (1 ref) #42
tv113-<*11> = CMP(NE) _j<6>, 8 #42
CBRANCH(NE) tv113-<11>, $L5, $L3 #42
==== Block 5 Pred(4) Succ(15,6) prev 4 next 6 pre 11 post 34 iDom 4 df 3,19
$L5: (1 ref) #42
tv114-<*12> = ADD _j<6>, 1 #44
_j<*7> = ASSIGN tv114-<12> #44
tv115-<*13> = ASSIGN _q<3> #45
[tv115-<13>] = ASSIGN 0 #45
tv116-<*14> = MUL _j<7>, 4 #46
tv117-<*15> = ADD _b<1>, tv116-<14> #46
tv119-<*16> = CMP(EQ) [tv117-<15>], 1 #46
CBRANCH(EQ) tv119-<16>, $L7, $L6 #46
==== Block 6 Pred(5) Succ(15,7) prev 5 next 7 pre 14 post 33 iDom 5 df 15,19
$L7: (1 ref) #46
tv120-<*17> = ADD _i<4>, _j<7> #46
tv121-<*18> = MUL tv120-<17>, 4 #46
tv122-<*19> = ADD _a<2>, tv121-<18> #46
tv124-<*20> = CMP(EQ) [tv122-<19>], 1 #46
CBRANCH(EQ) tv124-<20>, $L8, $L6 #46
==== Block 7 Pred(6) Succ(15,8) prev 6 next 8 pre 15 post 32 iDom 6 df 15,19
$L8: (1 ref) #46
tv126-<*21> = SUB _i<4>, _j<7> #46
tv127-<*22> = ADD tv126-<21>, 7 #46
tv128-<*23> = MUL tv127-<22>, 4 #46
tv129-<*24> = ADD &_c, tv128-<23> #46
tv131-<*25> = CMP(EQ) [tv129-<24>], 1 #46
CBRANCH(EQ) tv131-<25>, $L9, $L6 #46
==== Block 8 Pred(7) Succ(13,9) prev 7 next 9 pre 16 post 31 iDom 7 df 15,19
$L9: (1 ref) #46
tv133-<*26> = MUL _i<4>, 4 #48
tv134-<*27> = ADD &_x, tv133-<26> #48
[tv134-<27>] = ASSIGN _j<7> #48
tv136-<*28> = MUL _j<7>, 4 #49
tv137-<*29> = ADD _b<1>, tv136-<28> #49
[tv137-<29>] = ASSIGN 0 #49
tv139-<*30> = ADD _i<4>, _j<7> #50
tv140-<*31> = MUL tv139-<30>, 4 #50

```

AppendixF.txt

```

tv141-<*32> = ADD _a<2>, tv140-<31> #50
[tv141-<32>] = ASSIGN 0 #50
tv143-<*33> = SUB _i<4>, _j<7> #51
tv144-<*34> = ADD tv143-<33>, 7 #51
tv145-<*35> = MUL tv144-<34>, 4 #51
tv146-<*36> = ADD &_c, tv145-<35> #51
[tv146-<36>] = ASSIGN 0 #51
tv148-<*37> = CMP(LT) _i<4>, 8 #52
                CBRANCH(LT) tv148-<37>, $L11, $L10 #52
==== Block 9 Pred(8) Succ(10,17) prev 8 next 17 pre 21 post 30 iDom 8 df 14,
19
$L11: (1 ref) #52
        tv149-<*38> = ADD _i<4>, 1 #54
        {-5} = CALL _try, tv149-<38>, _q<3>, _a<2>, _b<1>, {-5} [Handler:
$L1]
#54
                GOTO $L16 #65
==== Block 17 Pred(9) Succ(19) prev 9 next 10 pre 28 post 29 iDom 9 df 19
$L1: (1 ref) #37
                UNWIND _try #37
==== Block 10 Pred(9) Succ(12,11) prev 17 next 11 pre 22 post 27 iDom 9 df 1
4
$L16: (1 ref) #65
        tv151-<*39> = ASSIGN _q<3> #55
        tv152-<*40> = CMP(EQ) [tv151-<39>], 0 #55
                CBRANCH(EQ) tv152-<40>, $L13, $L12 #55
==== Block 11 Pred(10) Succ(12) prev 10 next 12 pre 25 post 26 iDom 10 df 12
$L13: (1 ref) #55
        tv153-<*41> = MUL _j<7>, 4 #57
        tv154-<*42> = ADD _b<1>, tv153-<41> #57
        [tv154-<42>] = ASSIGN 1 #57
        tv156-<*43> = ADD _i<4>, _j<7> #58
        tv157-<*44> = MUL tv156-<43>, 4 #58
        tv158-<*45> = ADD _a<2>, tv157-<44> #58
        [tv158-<45>] = ASSIGN 1 #58
        tv160-<*46> = SUB _i<4>, _j<7> #59
        tv161-<*47> = ADD tv160-<46>, 7 #59
        tv162-<*48> = MUL tv161-<47>, 4 #59
        tv163-<*49> = ADD &_c, tv162-<48> #59
        [tv163-<49>] = ASSIGN 1 #59
                GOTO $L12 #55
==== Block 12 Pred(11,10) Succ(14) prev 11 next 13 pre 23 post 24 iDom 10 df
14
$L12: (2 refs) #55
                GOTO $L14 #62
==== Block 13 Pred(8) Succ(14) prev 12 next 14 pre 17 post 20 iDom 8 df 14
$L10: (1 ref) #52
        tv165-<*50> = ASSIGN _q<3> #62
        [tv165-<50>] = ASSIGN 1 #62
                GOTO $L14 #62
==== Block 14 Pred(13,12) Succ(15) prev 13 next 15 pre 18 post 19 iDom 8 df
15
$L14: (2 refs) #62

```


AppendixF.txt

```

                                GOTO $L6                                #46
==== Block 15 Pred(14,7,6,5) Succ(3) prev 14 next 16 pre 12 post 13 iDom 5 d
f 3
$L6:  (4 refs)                                #46
                                GOTO $L2                                #64
==== Block 16 Pred(4,3) Succ(18) prev 15 next 18 pre 4 post 9 iDom 3 df 19
$L3:  (2 refs)                                #42
                                GOTO $L15                                #65
==== Block 18 Pred(16) Succ(19) prev 16 next 19 pre 5 post 8 iDom 16 df 19
$L15: (1 ref)                                #65
                                EXITFUNC _try                            #65
==== Block 19 Pred(18,17) Succ() prev 18 pre 6 post 7 iDom 3 df
                                END _try, {-7}                            #65

```

IR after Ssa Info Destruction (flag Ssa)

```

{-7}          = START _try                                #37
_i<*4>, _q<*3>, _a<*2>, _b<*1> = ENTERFUNC _try            #37
_j<*5>        = ASSIGN 0                                    #40
t110          = ASSIGN _q                                    #41
[t110]        = ASSIGN 0                                    #41
                                GOTO $L2                                #42
$L2:  (2 refs)                                #42
t111          = ASSIGN _q                                    #42
t112          = CMP(EQ) [t111], 0                            #42
                                CBRANCH(EQ) t112, $L4, $L3            #42
$L4:  (1 ref)                                #42
t113          = CMP(NE) _j, 8                                #42
                                CBRANCH(NE) t113, $L5, $L3            #42
$L5:  (1 ref)                                #42
t114          = ADD _j, 1                                    #44
_j<*7>        = ASSIGN t114                                  #44
t115          = ASSIGN _q                                    #45
[t115]        = ASSIGN 0                                    #45
t116          = MUL _j, 4                                    #46
t117          = ADD _b, t116                                  #46
t119          = CMP(EQ) [t117], 1                            #46
                                CBRANCH(EQ) t119, $L7, $L6            #46
$L7:  (1 ref)                                #46
t120          = ADD _i, _j                                    #46
t121          = MUL t120, 4                                    #46
t122          = ADD _a, t121                                  #46
t124          = CMP(EQ) [t122], 1                            #46
                                CBRANCH(EQ) t124, $L8, $L6            #46
$L8:  (1 ref)                                #46
t126          = SUB _i, _j                                    #46
t127          = ADD t126, 7                                    #46
t128          = MUL t127, 4                                    #46
t129          = ADD &_c, t128                                  #46
t131          = CMP(EQ) [t129], 1                            #46
                                CBRANCH(EQ) t131, $L9, $L6            #46
$L9:  (1 ref)                                #46

```

t133	= MUL _i, 4	#48
t134	= ADD &_x, t133	#48
[t134]	= ASSIGN _j	#48
t136	= MUL _j, 4	#49
t137	= ADD _b, t136	#49
[t137]	= ASSIGN 0	#49
t139	= ADD _i, _j	#50
t140	= MUL t139, 4	#50
t141	= ADD _a, t140	#50
[t141]	= ASSIGN 0	#50
t143	= SUB _i, _j	#51
t144	= ADD t143, 7	#51
t145	= MUL t144, 4	#51
t146	= ADD &_c, t145	#51
[t146]	= ASSIGN 0	#51
t148	= CMP(LT) _i, 8	#52
	CBRANCH(LT) t148, \$L11, \$L10	#52
\$L11: (1 ref)		#52
t149	= ADD _i, 1	#54
{-5}	= CALL _try, t149, _q, _a, _b, {-5} [Handler: \$L1]	#54
	GOTO \$L16	#65
\$L16: (1 ref)		#65
t151	= ASSIGN _q	#55
t152	= CMP(EQ) [t151], 0	#55
	CBRANCH(EQ) t152, \$L13, \$L12	#55
\$L13: (1 ref)		#55
t153	= MUL _j, 4	#57
t154	= ADD _b, t153	#57
[t154]	= ASSIGN 1	#57
t156	= ADD _i, _j	#58
t157	= MUL t156, 4	#58
t158	= ADD _a, t157	#58
[t158]	= ASSIGN 1	#58
t160	= SUB _i, _j	#59
t161	= ADD t160, 7	#59
t162	= MUL t161, 4	#59
t163	= ADD &_c, t162	#59
[t163]	= ASSIGN 1	#59
	GOTO \$L12	#55
\$L12: (2 refs)		#55
	GOTO \$L14	#62
\$L10: (1 ref)		#52
t165	= ASSIGN _q	#62
[t165]	= ASSIGN 1	#62
	GOTO \$L14	#62
\$L14: (2 refs)		#62
	GOTO \$L6	#46
\$L6: (4 refs)		#46
	GOTO \$L2	#64
\$L3: (2 refs)		#42
	GOTO \$L15	#65
\$L1: (1 ref)		#37
	UNWIND _try	#37

```

$15:  (1 ref)                                     #65
        EXITFUNC _try                             #65
        END _try, {-7}                           #65

IR after Lower (flag Lower)

        {-7} = START _try                         #37
        _i<*4>, _q<*3>, _a<*2>, _b<*1> = ENTERFUNC _try #37
        PROLOGEND                                #37
        _j[_FP] = mov 0                           #40
        t110(_rd) = mov _q[_FP]                   #41
        [t110(_rd)] = mov 0                       #41
        jmp $L2                                    #42
$L2:  (2 refs)                                    #42
        t111(_rd) = mov _q[_FP]                   #42
        t112(EFLAGS) = cmp(EQ) [t111(_rd)], 0      #42
        jcc(EQ) t112(EFLAGS), $L4, $L3            #42
$L4:  (1 ref)                                    #42
        t113(EFLAGS) = cmp(NE) _j[_FP], 8          #42
        jcc(NE) t113(EFLAGS), $L5, $L3            #42
$L5:  (1 ref)                                    #42
        tv114-(_rd) = mov 1                       #44
        tv114-(_rd)<*12>, EFLAGS = add tv114-(_rd), _j[_FP] #44
        _j[_FP] = mov tv114-(_rd)                 #44
        t115(_rd) = mov _q[_FP]                   #45
        [t115(_rd)] = mov 0                       #45
        t116(_rd), EFLAGS = imul _j[_FP], 4        #46
        tv117-(_rd) = mov t116(_rd)               #46
        tv117-(_rd)<*15>, EFLAGS = add tv117-(_rd), _b[_FP] #46
        t119(EFLAGS) = cmp(EQ) [tv117-(_rd)], 1    #46
        jcc(EQ) t119(EFLAGS), $L7, $L6            #46
$L7:  (1 ref)                                    #46
        tv120-(_rd) = mov _i[_FP]                 #46
        tv120-(_rd)<*17>, EFLAGS = add tv120-(_rd), _j[_FP] #46
        t121(_rd), EFLAGS = imul tv120-(_rd), 4    #46
        tv122-(_rd) = mov t121(_rd)               #46
        tv122-(_rd)<*19>, EFLAGS = add tv122-(_rd), _a[_FP] #46
        t124(EFLAGS) = cmp(EQ) [tv122-(_rd)], 1    #46
        jcc(EQ) t124(EFLAGS), $L8, $L6            #46
$L8:  (1 ref)                                    #46
        tv126-(_rd) = mov _i[_FP]                 #46
        tv126-(_rd)<*21>, EFLAGS = sub tv126-(_rd), _j[_FP] #46
        tv127-(_rd) = mov tv126-(_rd)             #46
        tv127-(_rd)<*22>, EFLAGS = add tv127-(_rd), 7 #46
        t128(_rd), EFLAGS = imul tv127-(_rd), 4    #46
        tv129-(_rd) = mov t128(_rd)               #46
        tv129-(_rd)<*24>, EFLAGS = add tv129-(_rd), &_c #46
        t131(EFLAGS) = cmp(EQ) [tv129-(_rd)], 1    #46
        jcc(EQ) t131(EFLAGS), $L9, $L6            #46
$L9:  (1 ref)                                    #46
        t133(_rd), EFLAGS = imul _i[_FP], 4        #48
        tv134-(_rd) = mov t133(_rd)               #48

```

AppendixF.txt

```

tv134-(_rd)<*27>, EFLAGS = add tv134-(_rd), &_x      #48
t169(_rd) = mov _j[_FP]                               #48
[tv134-(_rd)] = mov t169(_rd)                         #48
t136(_rd), EFLAGS = imul _j[_FP], 4                   #49
tv137-(_rd) = mov t136(_rd)                           #49
tv137-(_rd)<*29>, EFLAGS = add tv137-(_rd), _b[_FP]    #49
[tv137-(_rd)] = mov 0                                #49
tv139-(_rd) = mov _i[_FP]                             #50
tv139-(_rd)<*30>, EFLAGS = add tv139-(_rd), _j[_FP]    #50
t140(_rd), EFLAGS = imul tv139-(_rd), 4               #50
tv141-(_rd) = mov t140(_rd)                           #50
tv141-(_rd)<*32>, EFLAGS = add tv141-(_rd), _a[_FP]    #50
[tv141-(_rd)] = mov 0                                #50
tv143-(_rd) = mov _i[_FP]                             #51
tv143-(_rd)<*33>, EFLAGS = sub tv143-(_rd), _j[_FP]    #51
tv144-(_rd) = mov tv143-(_rd)                         #51
tv144-(_rd)<*34>, EFLAGS = add tv144-(_rd), 7          #51
t145(_rd), EFLAGS = imul tv144-(_rd), 4               #51
tv146-(_rd) = mov t145(_rd)                           #51
tv146-(_rd)<*36>, EFLAGS = add tv146-(_rd), &_c       #51
[tv146-(_rd)] = mov 0                                #51
t148(EFLAGS) = cmp(LT) _i[_FP], 8                     #52
                jcc(LT) t148(EFLAGS), $L11, $L10       #52
$L11: (1 ref)                                          #52
    tv149-(_rd) = mov 1                               #54
    tv149-(_rd)<*38>, EFLAGS = add tv149-(_rd), _i[_FP] #54
    [ESP], {ESP} = push _b[_FP], {ESP}                 #54
    [ESP], {ESP} = push _a[_FP], {ESP}                 #54
    [ESP], {ESP} = push _q[_FP], {ESP}                 #54
    [ESP], {ESP} = push tv149-(_rd), {ESP}             #54
    {-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7 FPUStatus} = call
try, $out[ESP], $out[ESP]+32, $out[ESP]+64, $out[ESP]+96, {-5}, {EAX ECX E
DX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7 FPUStatus} [Handler: $L1]
    #54
    ESP, EFLAGS = add ESP, 16(0x00000010)              #54
                jmp $L16                               #65
$L16: (1 ref)                                          #65
    t151(_rd) = mov _q[_FP]                             #55
    t152(EFLAGS) = cmp(EQ) [t151(_rd)], 0              #55
                jcc(EQ) t152(EFLAGS), $L13, $L12       #55
$L13: (1 ref)                                          #55
    t153(_rd), EFLAGS = imul _j[_FP], 4                #57
    tv154-(_rd) = mov t153(_rd)                         #57
    tv154-(_rd)<*42>, EFLAGS = add tv154-(_rd), _b[_FP] #57
    [tv154-(_rd)] = mov 1                               #57
    tv156-(_rd) = mov _i[_FP]                             #58
    tv156-(_rd)<*43>, EFLAGS = add tv156-(_rd), _j[_FP] #58
    t157(_rd), EFLAGS = imul tv156-(_rd), 4            #58
    tv158-(_rd) = mov t157(_rd)                         #58
    tv158-(_rd)<*45>, EFLAGS = add tv158-(_rd), _a[_FP] #58
    [tv158-(_rd)] = mov 1                               #58
    tv160-(_rd) = mov _i[_FP]                             #59
    tv160-(_rd)<*46>, EFLAGS = sub tv160-(_rd), _j[_FP] #59

```

AppendixF.txt

```

tv161-(_rd) = mov tv160-(_rd) #59
tv161-(_rd)<*47>, EFLAGS = add tv161-(_rd), 7 #59
t162(_rd), EFLAGS = imul tv161-(_rd), 4 #59
tv163-(_rd) = mov t162(_rd) #59
tv163-(_rd)<*49>, EFLAGS = add tv163-(_rd), &_c #59
[tv163-(_rd)] = mov 1 #59
        jmp $L12 #55
$L12: (2 refs) #55
        jmp $L14 #62
$L10: (1 ref) #52
        t165(_rd) = mov _q[_FP] #62
        [t165(_rd)] = mov 1 #62
        jmp $L14 #62
$L14: (2 refs) #62
        jmp $L6 #46
$L6: (4 refs) #46
        jmp $L2 #64
$L3: (2 refs) #42
        jmp $L15 #65
$L1: (1 ref) #37
        UNWIND _try #37
$L15: (1 ref) #65
        EPILOGSTART #65
        EXITFUNC _try #65
        END _try, {-7} #65

```

IR after Linear Scan Register Allocation (flag LinearScan)

```

{-7} = START _try #37
_i<*4>, _q<*3>, _a<*2>, _b<*1> = ENTERFUNC _try #37
        PROLOGEND #37
        _j[_FP] = mov 0 #40
        t110(EAX) = mov _q[_FP] #41
        [t110(EAX)] = mov 0 #41
        jmp $L2 #42
$L2: (2 refs) #42
        t111(EAX) = mov _q[_FP] #42
        t112(EFLAGS) = cmp(EQ) [t111(EAX)], 0 #42
        jcc(EQ) t112(EFLAGS), $L4, $L3 #42
$L4: (1 ref) #42
        t113(EFLAGS) = cmp(NE) _j[_FP], 8 #42
        jcc(NE) t113(EFLAGS), $L5, $L3 #42
$L5: (1 ref) #42
        tv114-(EAX) = mov 1 #44
        tv114-(EAX)<*12>, EFLAGS = add tv114-(EAX), _j[_FP] #44
        _j[_FP] = mov tv114-(EAX) #44
        t115(EAX) = mov _q[_FP] #45
        [t115(EAX)] = mov 0 #45
        t116(EAX), EFLAGS = imul _j[_FP], 4 #46
        tv117-(EAX) = mov t116(EAX) #46
        tv117-(EAX)<*15>, EFLAGS = add tv117-(EAX), _b[_FP] #46
        t119(EFLAGS) = cmp(EQ) [tv117-(EAX)], 1 #46

```

AppendixF.txt

```

                                jcc(EQ) t119(EFLAGS), $L7, $L6                                #46
$ L7: (1 ref)                                                            #46
    tv120-(EAX) = mov _i[_FP]                                           #46
    tv120-(EAX)<*17>, EFLAGS = add tv120-(EAX), _j[_FP]                 #46
    t121(EAX), EFLAGS = imul tv120-(EAX), 4                             #46
    tv122-(EAX) = mov t121(EAX)                                         #46
    tv122-(EAX)<*19>, EFLAGS = add tv122-(EAX), _a[_FP]                 #46
    t124(EFLAGS) = cmp(EQ) [tv122-(EAX)], 1                             #46
                                jcc(EQ) t124(EFLAGS), $L8, $L6          #46
$ L8: (1 ref)                                                            #46
    tv126-(EAX) = mov _i[_FP]                                           #46
    tv126-(EAX)<*21>, EFLAGS = sub tv126-(EAX), _j[_FP]                 #46
    tv127-(EAX) = mov tv126-(EAX)                                       #46
    tv127-(EAX)<*22>, EFLAGS = add tv127-(EAX), 7                       #46
    t128(EAX), EFLAGS = imul tv127-(EAX), 4                             #46
    tv129-(EAX) = mov t128(EAX)                                         #46
    tv129-(EAX)<*24>, EFLAGS = add tv129-(EAX), &_amp;c                 #46
    t131(EFLAGS) = cmp(EQ) [tv129-(EAX)], 1                             #46
                                jcc(EQ) t131(EFLAGS), $L9, $L6          #46
$ L9: (1 ref)                                                            #46
    t133(EAX), EFLAGS = imul _i[_FP], 4                                 #48
    tv134-(EAX) = mov t133(EAX)                                         #48
    tv134-(EAX)<*27>, EFLAGS = add tv134-(EAX), &_amp;x                 #48
    t169(ECX) = mov _j[_FP]                                             #48
    [tv134-(EAX)] = mov t169(ECX)                                       #48
    t136(EAX), EFLAGS = imul _j[_FP], 4                                 #49
    tv137-(EAX) = mov t136(EAX)                                         #49
    tv137-(EAX)<*29>, EFLAGS = add tv137-(EAX), _b[_FP]                 #49
    [tv137-(EAX)] = mov 0                                              #49
    tv139-(EAX) = mov _i[_FP]                                           #50
    tv139-(EAX)<*30>, EFLAGS = add tv139-(EAX), _j[_FP]                 #50
    t140(EAX), EFLAGS = imul tv139-(EAX), 4                             #50
    tv141-(EAX) = mov t140(EAX)                                         #50
    tv141-(EAX)<*32>, EFLAGS = add tv141-(EAX), _a[_FP]                 #50
    [tv141-(EAX)] = mov 0                                              #50
    tv143-(EAX) = mov _i[_FP]                                           #51
    tv143-(EAX)<*33>, EFLAGS = sub tv143-(EAX), _j[_FP]                 #51
    tv144-(EAX) = mov tv143-(EAX)                                       #51
    tv144-(EAX)<*34>, EFLAGS = add tv144-(EAX), 7                       #51
    t145(EAX), EFLAGS = imul tv144-(EAX), 4                             #51
    tv146-(EAX) = mov t145(EAX)                                         #51
    tv146-(EAX)<*36>, EFLAGS = add tv146-(EAX), &_amp;c                 #51
    [tv146-(EAX)] = mov 0                                              #51
    t148(EFLAGS) = cmp(LT) _i[_FP], 8                                   #52
                                jcc(LT) t148(EFLAGS), $L11, $L10        #52
$ L11: (1 ref)                                                            #52
    tv149-(EAX) = mov 1                                                 #54
    tv149-(EAX)<*38>, EFLAGS = add tv149-(EAX), _i[_FP]                 #54
    [ESP], {ESP} = push _b[_FP], {ESP}                                  #54
    [ESP], {ESP} = push _a[_FP], {ESP}                                  #54
    [ESP], {ESP} = push _q[_FP], {ESP}                                  #54
    [ESP], {ESP} = push tv149-(EAX), {ESP}                              #54
    {-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7 FPUSatus} = call

```

AppendixF.txt

```

_try, $out[ESP], $out[ESP]+32, $out[ESP]+64, $out[ESP]+96, {-5}, {EAX ECX E
DX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7 FPUStatus} [Handler: $L1]

```

```

#54
        jmp $L17                                #65
$L17:   (1 ref)                                #65
        ESP, EFLAGS = add ESP, 16(0x00000010)   #54
        jmp $L16                                #65
$L16:   (1 ref)                                #65
        t151(EAX) = mov _q[_FP]                 #55
        t152(EFLAGS) = cmp(EQ) [t151(EAX)], 0    #55
        jcc(EQ) t152(EFLAGS), $L13, $L12        #55
$L13:   (1 ref)                                #55
        t153(EAX), EFLAGS = imul _j[_FP], 4      #57
        tv154-(EAX) = mov t153(EAX)             #57
        tv154-(EAX)<*42>, EFLAGS = add tv154-(EAX), _b[_FP] #57
        [tv154-(EAX)] = mov 1                   #57
        tv156-(EAX) = mov _i[_FP]               #58
        tv156-(EAX)<*43>, EFLAGS = add tv156-(EAX), _j[_FP] #58
        t157(EAX), EFLAGS = imul tv156-(EAX), 4  #58
        tv158-(EAX) = mov t157(EAX)             #58
        tv158-(EAX)<*45>, EFLAGS = add tv158-(EAX), _a[_FP] #58
        [tv158-(EAX)] = mov 1                   #58
        tv160-(EAX) = mov _i[_FP]               #59
        tv160-(EAX)<*46>, EFLAGS = sub tv160-(EAX), _j[_FP] #59
        tv161-(EAX) = mov tv160-(EAX)           #59
        tv161-(EAX)<*47>, EFLAGS = add tv161-(EAX), 7  #59
        t162(EAX), EFLAGS = imul tv161-(EAX), 4    #59
        tv163-(EAX) = mov t162(EAX)             #59
        tv163-(EAX)<*49>, EFLAGS = add tv163-(EAX), &_c #59
        [tv163-(EAX)] = mov 1                   #59
        jmp $L12                                #55
$L12:   (2 refs)                                #55
        jmp $L14                                #62
$L10:   (1 ref)                                #52
        t165(EAX) = mov _q[_FP]                 #62
        [t165(EAX)] = mov 1                     #62
        jmp $L14                                #62
$L14:   (2 refs)                                #62
        jmp $L6                                  #46
$L6:    (4 refs)                                #46
        jmp $L2                                  #64
$L3:    (2 refs)                                #42
        jmp $L15                                #65
$L1:    (1 ref)                                #37
        UNWIND _try                             #37
$L15:   (1 ref)                                #65
        EPILOGSTART                             #65
        EXITFUNC _try                          #65
        END _try, {-7}                         #65

```

IR after Stack Allocation (flag StackAlloc)

AppendixF.txt

```

{-7}          = START _try                                #37
_i<*4>, _q<*3>, _a<*2>, _b<*1> = ENTERFUNC _try           #37
        PROLOGEND                                         #37
_j[_FP]       = mov 0                                     #40
t110(EAX)     = mov _q[_FP]                               #41
[t110(EAX)]   = mov 0                                     #41
        jmp $L2                                           #42
$L2: (2 refs)                                           #42
t111(EAX)     = mov _q[_FP]                               #42
t112(EFLAGS) = cmp(EQ) [t111(EAX)], 0                     #42
        jcc(EQ) t112(EFLAGS), $L4, $L3                    #42
$L4: (1 ref)                                           #42
t113(EFLAGS) = cmp(NE) _j[_FP], 8                         #42
        jcc(NE) t113(EFLAGS), $L5, $L3                    #42
$L5: (1 ref)                                           #42
tv114-(EAX) = mov 1                                       #44
tv114-(EAX)<*12>, EFLAGS = add tv114-(EAX), _j[_FP]       #44
_j[_FP]      = mov tv114-(EAX)                           #44
t115(EAX)    = mov _q[_FP]                               #45
[t115(EAX)]  = mov 0                                     #45
t116(EAX), EFLAGS = imul _j[_FP], 4                      #46
tv117-(EAX) = mov t116(EAX)                              #46
tv117-(EAX)<*15>, EFLAGS = add tv117-(EAX), _b[_FP]       #46
t119(EFLAGS) = cmp(EQ) [tv117-(EAX)], 1                  #46
        jcc(EQ) t119(EFLAGS), $L7, $L6                    #46
$L7: (1 ref)                                           #46
tv120-(EAX) = mov _i[_FP]                                #46
tv120-(EAX)<*17>, EFLAGS = add tv120-(EAX), _j[_FP]       #46
t121(EAX), EFLAGS = imul tv120-(EAX), 4                  #46
tv122-(EAX) = mov t121(EAX)                              #46
tv122-(EAX)<*19>, EFLAGS = add tv122-(EAX), _a[_FP]       #46
t124(EFLAGS) = cmp(EQ) [tv122-(EAX)], 1                  #46
        jcc(EQ) t124(EFLAGS), $L8, $L6                    #46
$L8: (1 ref)                                           #46
tv126-(EAX) = mov _i[_FP]                                #46
tv126-(EAX)<*21>, EFLAGS = sub tv126-(EAX), _j[_FP]       #46
tv127-(EAX) = mov tv126-(EAX)                            #46
tv127-(EAX)<*22>, EFLAGS = add tv127-(EAX), 7             #46
t128(EAX), EFLAGS = imul tv127-(EAX), 4                  #46
tv129-(EAX) = mov t128(EAX)                              #46
tv129-(EAX)<*24>, EFLAGS = add tv129-(EAX), &_c           #46
t131(EFLAGS) = cmp(EQ) [tv129-(EAX)], 1                  #46
        jcc(EQ) t131(EFLAGS), $L9, $L6                    #46
$L9: (1 ref)                                           #46
t133(EAX), EFLAGS = imul _i[_FP], 4                      #48
tv134-(EAX) = mov t133(EAX)                              #48
tv134-(EAX)<*27>, EFLAGS = add tv134-(EAX), &_x           #48
t169(ECX)    = mov _j[_FP]                               #48
[tv134-(EAX)] = mov t169(ECX)                            #48
t136(EAX), EFLAGS = imul _j[_FP], 4                      #49
tv137-(EAX) = mov t136(EAX)                              #49
tv137-(EAX)<*29>, EFLAGS = add tv137-(EAX), _b[_FP]       #49
[tv137-(EAX)] = mov 0                                     #49

```


AppendixF.txt

```

tv139-(EAX) = mov _i[_FP] #50
tv139-(EAX)<*30>, EFLAGS = add tv139-(EAX), _j[_FP] #50
t140(EAX), EFLAGS = imul tv139-(EAX), 4 #50
tv141-(EAX) = mov t140(EAX) #50
tv141-(EAX)<*32>, EFLAGS = add tv141-(EAX), _a[_FP] #50
[tv141-(EAX)] = mov 0 #50
tv143-(EAX) = mov _i[_FP] #51
tv143-(EAX)<*33>, EFLAGS = sub tv143-(EAX), _j[_FP] #51
tv144-(EAX) = mov tv143-(EAX) #51
tv144-(EAX)<*34>, EFLAGS = add tv144-(EAX), 7 #51
t145(EAX), EFLAGS = imul tv144-(EAX), 4 #51
tv146-(EAX) = mov t145(EAX) #51
tv146-(EAX)<*36>, EFLAGS = add tv146-(EAX), &_c #51
[tv146-(EAX)] = mov 0 #51
t148(EFLAGS) = cmp(LT) _i[_FP], 8 #52
jcc(LT) t148(EFLAGS), $L11, $L10 #52
$L11: (1 ref) #52
tv149-(EAX) = mov 1 #54
tv149-(EAX)<*38>, EFLAGS = add tv149-(EAX), _i[_FP] #54
[ESP], {ESP} = push _b[_FP], {ESP} #54
[ESP], {ESP} = push _a[_FP], {ESP} #54
[ESP], {ESP} = push _q[_FP], {ESP} #54
[ESP], {ESP} = push tv149-(EAX), {ESP} #54
{-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7 FPUStatus} = call
try, $out[ESP], $out[ESP]+32, $out[ESP]+64, $out[ESP]+96, {-5}, {EAX ECX E
DX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7 FPUStatus} [Handler: $L1]
#54
jmp $L17 #65
$L17: (1 ref) #65
ESP, EFLAGS = add ESP, 16(0x00000010) #54
jmp $L16 #65
$L16: (1 ref) #65
t151(EAX) = mov _q[_FP] #55
t152(EFLAGS) = cmp(EQ) [t151(EAX)], 0 #55
jcc(EQ) t152(EFLAGS), $L13, $L12 #55
$L13: (1 ref) #55
t153(EAX), EFLAGS = imul _j[_FP], 4 #57
tv154-(EAX) = mov t153(EAX) #57
tv154-(EAX)<*42>, EFLAGS = add tv154-(EAX), _b[_FP] #57
[tv154-(EAX)] = mov 1 #57
tv156-(EAX) = mov _i[_FP] #58
tv156-(EAX)<*43>, EFLAGS = add tv156-(EAX), _j[_FP] #58
t157(EAX), EFLAGS = imul tv156-(EAX), 4 #58
tv158-(EAX) = mov t157(EAX) #58
tv158-(EAX)<*45>, EFLAGS = add tv158-(EAX), _a[_FP] #58
[tv158-(EAX)] = mov 1 #58
tv160-(EAX) = mov _i[_FP] #59
tv160-(EAX)<*46>, EFLAGS = sub tv160-(EAX), _j[_FP] #59
tv161-(EAX) = mov tv160-(EAX) #59
tv161-(EAX)<*47>, EFLAGS = add tv161-(EAX), 7 #59
t162(EAX), EFLAGS = imul tv161-(EAX), 4 #59
tv163-(EAX) = mov t162(EAX) #59
tv163-(EAX)<*49>, EFLAGS = add tv163-(EAX), &_c #59

```

[tv163-(EAX)] = mov 1	#59
jmp \$L12	#55
\$L12: (2 refs)	#55
jmp \$L14	#62
\$L10: (1 ref)	#52
t165(EAX) = mov _q[_FP]	#62
[t165(EAX)] = mov 1	#62
jmp \$L14	#62
\$L14: (2 refs)	#62
jmp \$L6	#46
\$L6: (4 refs)	#46
jmp \$L2	#64
\$L3: (2 refs)	#42
jmp \$L15	#65
\$L1: (1 ref)	#37
UNWIND _try	#37
\$L15: (1 ref)	#65
EPILOGSTART	#65
EXITFUNC _try	#65
END _try, {-7}	#65

IR after Frame Generation (flag Frame)

{-7}	= START _try	#37
_i<*4>, _q<*3>, _a<*2>, _b<*1>	= ENTERFUNC _try	#37
[ESP], {ESP}	= push EBP, {ESP}	#37
EBP	= mov ESP	#37
[ESP], {ESP}	= push ESP, {ESP}	#37
	PROLOGEND	#37
_j[EBP]	= mov 0	#40
t110(EAX)	= mov _q[EBP]	#41
[t110(EAX)]	= mov 0	#41
	jmp \$L2	#42
\$L2: (2 refs)		#42
t111(EAX)	= mov _q[EBP]	#42
t112(EFLAGS)	= cmp(EQ) [t111(EAX)], 0	#42
	jcc(EQ) t112(EFLAGS), \$L4, \$L3	#42
\$L4: (1 ref)		#42
t113(EFLAGS)	= cmp(NE) _j[EBP], 8	#42
	jcc(NE) t113(EFLAGS), \$L5, \$L3	#42
\$L5: (1 ref)		#42
tv114-(EAX)	= mov 1	#44
tv114-(EAX)<*12>, EFLAGS	= add tv114-(EAX), _j[EBP]	#44
_j[EBP]	= mov tv114-(EAX)	#44
t115(EAX)	= mov _q[EBP]	#45
[t115(EAX)]	= mov 0	#45
t116(EAX), EFLAGS	= imul _j[EBP], 4	#46
tv117-(EAX)	= mov t116(EAX)	#46
tv117-(EAX)<*15>, EFLAGS	= add tv117-(EAX), _b[EBP]	#46
t119(EFLAGS)	= cmp(EQ) [tv117-(EAX)], 1	#46
	jcc(EQ) t119(EFLAGS), \$L7, \$L6	#46
\$L7: (1 ref)		#46

AppendixF.txt

```

tv120-(EAX) = mov _i[EBP] #46
tv120-(EAX)<*17>, EFLAGS = add tv120-(EAX), _j[EBP] #46
t121(EAX), EFLAGS = imul tv120-(EAX), 4 #46
tv122-(EAX) = mov t121(EAX) #46
tv122-(EAX)<*19>, EFLAGS = add tv122-(EAX), _a[EBP] #46
t124(EFLAGS) = cmp(EQ) [tv122-(EAX)], 1 #46
jcc(EQ) t124(EFLAGS), $L8, $L6 #46
$L8: (1 ref) #46
tv126-(EAX) = mov _i[EBP] #46
tv126-(EAX)<*21>, EFLAGS = sub tv126-(EAX), _j[EBP] #46
tv127-(EAX) = mov tv126-(EAX) #46
tv127-(EAX)<*22>, EFLAGS = add tv127-(EAX), 7 #46
t128(EAX), EFLAGS = imul tv127-(EAX), 4 #46
tv129-(EAX) = mov t128(EAX) #46
tv129-(EAX)<*24>, EFLAGS = add tv129-(EAX), &_c #46
t131(EFLAGS) = cmp(EQ) [tv129-(EAX)], 1 #46
jcc(EQ) t131(EFLAGS), $L9, $L6 #46
$L9: (1 ref) #46
t133(EAX), EFLAGS = imul _i[EBP], 4 #48
tv134-(EAX) = mov t133(EAX) #48
tv134-(EAX)<*27>, EFLAGS = add tv134-(EAX), &_x #48
t169(ECX) = mov _j[EBP] #48
[tv134-(EAX)] = mov t169(ECX) #48
t136(EAX), EFLAGS = imul _j[EBP], 4 #49
tv137-(EAX) = mov t136(EAX) #49
tv137-(EAX)<*29>, EFLAGS = add tv137-(EAX), _b[EBP] #49
[tv137-(EAX)] = mov 0 #49
tv139-(EAX) = mov _i[EBP] #50
tv139-(EAX)<*30>, EFLAGS = add tv139-(EAX), _j[EBP] #50
t140(EAX), EFLAGS = imul tv139-(EAX), 4 #50
tv141-(EAX) = mov t140(EAX) #50
tv141-(EAX)<*32>, EFLAGS = add tv141-(EAX), _a[EBP] #50
[tv141-(EAX)] = mov 0 #50
tv143-(EAX) = mov _i[EBP] #51
tv143-(EAX)<*33>, EFLAGS = sub tv143-(EAX), _j[EBP] #51
tv144-(EAX) = mov tv143-(EAX) #51
tv144-(EAX)<*34>, EFLAGS = add tv144-(EAX), 7 #51
t145(EAX), EFLAGS = imul tv144-(EAX), 4 #51
tv146-(EAX) = mov t145(EAX) #51
tv146-(EAX)<*36>, EFLAGS = add tv146-(EAX), &_c #51
[tv146-(EAX)] = mov 0 #51
t148(EFLAGS) = cmp(LT) _i[EBP], 8 #52
jcc(LT) t148(EFLAGS), $L11, $L10 #52
$L11: (1 ref) #52
tv149-(EAX) = mov 1 #54
tv149-(EAX)<*38>, EFLAGS = add tv149-(EAX), _i[EBP] #54
[ESP], {ESP} = push _b[EBP], {ESP} #54
[ESP], {ESP} = push _a[EBP], {ESP} #54
[ESP], {ESP} = push _q[EBP], {ESP} #54
[ESP], {ESP} = push tv149-(EAX), {ESP} #54
{-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7 FPUStatus} = call
try, $out[ESP], $out[ESP]+32, $out[ESP]+64, $out[ESP]+96, {-5}, {EAX ECX E
DX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7 FPUStatus} [Handler: $L1]

```

```

#54                                #65
                                jmp $L17                                #65
$L17: (1 ref)                      #54
    ESP, EFLAGS = add ESP, 16(0x00000010) #65
                                jmp $L16                                #65
$L16: (1 ref)                      #65
    t151(EAX) = mov _q[EBP]        #55
    t152(EFLAGS) = cmp(EQ) [t151(EAX)], 0 #55
                                jcc(EQ) t152(EFLAGS), $L13, $L12 #55
$L13: (1 ref)                      #55
    t153(EAX), EFLAGS = imul _j[EBP], 4 #57
    tv154-(EAX) = mov t153(EAX)    #57
    tv154-(EAX)<*42>, EFLAGS = add tv154-(EAX), _b[EBP] #57
    [tv154-(EAX)] = mov 1          #57
    tv156-(EAX) = mov _i[EBP]     #58
    tv156-(EAX)<*43>, EFLAGS = add tv156-(EAX), _j[EBP] #58
    t157(EAX), EFLAGS = imul tv156-(EAX), 4 #58
    tv158-(EAX) = mov t157(EAX)   #58
    tv158-(EAX)<*45>, EFLAGS = add tv158-(EAX), _a[EBP] #58
    [tv158-(EAX)] = mov 1          #58
    tv160-(EAX) = mov _i[EBP]     #59
    tv160-(EAX)<*46>, EFLAGS = sub tv160-(EAX), _j[EBP] #59
    tv161-(EAX) = mov tv160-(EAX) #59
    tv161-(EAX)<*47>, EFLAGS = add tv161-(EAX), 7 #59
    t162(EAX), EFLAGS = imul tv161-(EAX), 4 #59
    tv163-(EAX) = mov t162(EAX)   #59
    tv163-(EAX)<*49>, EFLAGS = add tv163-(EAX), &_c #59
    [tv163-(EAX)] = mov 1          #59
                                jmp $L12                                #55
$L12: (2 refs)                      #55
                                jmp $L14                                #62
$L10: (1 ref)                      #52
    t165(EAX) = mov _q[EBP]        #62
    [t165(EAX)] = mov 1            #62
                                jmp $L14                                #62
$L14: (2 refs)                      #62
                                jmp $L6                                #46
$L6: (4 refs)                      #46
                                jmp $L2                                #64
$L3: (2 refs)                      #42
                                jmp $L15                                #65
$L1: (1 ref)                      #37
                                UNWIND _try                            #37
$L15: (1 ref)                      #65
                                EPILOGSTART                            #65
    ESP = mov EBP                #65
    EBP, {ESP} = pop [ESP], {ESP} #65
    {ESP} = ret {ESP}            #65
                                EXITFUNC _try                            #65
                                END _try, {-7}                            #65

```

AppendixF.txt

```

{-7}          = START _try                                #37
_i<*4>, _q<*3>, _a<*2>, _b<*1> = ENTERFUNC _try           #37
[ESP], {ESP} = push EBP, {ESP}                            #37
EBP           = mov ESP                                    #37
[ESP], {ESP} = push ESP, {ESP}                            #37
               PROLOGEND                                  #37
_j[EBP]       = mov 0                                     #40
t110(EAX)     = mov _q[EBP]                              #41
[t110(EAX)]   = mov 0                                     #41
               jmp $L2                                    #42
$L2: (2 refs)                                           #42
t111(EAX)     = mov _q[EBP]                              #42
t112(EFLAGS) = cmp(EQ) [t111(EAX)], 0                    #42
               jcc(EQ) t112(EFLAGS), $L4, $L3             #42
$L4: (1 ref)                                           #42
t113(EFLAGS) = cmp(NE) _j[EBP], 8                        #42
               jcc(NE) t113(EFLAGS), $L5, $L3             #42
$L5: (1 ref)                                           #42
tv114-(EAX) = mov 1                                       #44
tv114-(EAX)<*12>, EFLAGS = add tv114-(EAX), _j[EBP]      #44
_j[EBP]      = mov tv114-(EAX)                           #44
t115(EAX)    = mov _q[EBP]                              #45
[t115(EAX)]  = mov 0                                     #45
t116(EAX), EFLAGS = imul _j[EBP], 4                     #46
tv117-(EAX) = mov t116(EAX)                             #46
tv117-(EAX)<*15>, EFLAGS = add tv117-(EAX), _b[EBP]      #46
t119(EFLAGS) = cmp(EQ) [tv117-(EAX)], 1                  #46
               jcc(EQ) t119(EFLAGS), $L7, $L6             #46
$L7: (1 ref)                                           #46
tv120-(EAX) = mov _i[EBP]                                #46
tv120-(EAX)<*17>, EFLAGS = add tv120-(EAX), _j[EBP]      #46
t121(EAX), EFLAGS = imul tv120-(EAX), 4                  #46
tv122-(EAX) = mov t121(EAX)                             #46
tv122-(EAX)<*19>, EFLAGS = add tv122-(EAX), _a[EBP]      #46
t124(EFLAGS) = cmp(EQ) [tv122-(EAX)], 1                  #46
               jcc(EQ) t124(EFLAGS), $L8, $L6             #46
$L8: (1 ref)                                           #46
tv126-(EAX) = mov _i[EBP]                                #46
tv126-(EAX)<*21>, EFLAGS = sub tv126-(EAX), _j[EBP]      #46
tv127-(EAX) = mov tv126-(EAX)                           #46
tv127-(EAX)<*22>, EFLAGS = add tv127-(EAX), 7            #46
t128(EAX), EFLAGS = imul tv127-(EAX), 4                  #46
tv129-(EAX) = mov t128(EAX)                             #46
tv129-(EAX)<*24>, EFLAGS = add tv129-(EAX), &_amp;c        #46
t131(EFLAGS) = cmp(EQ) [tv129-(EAX)], 1                  #46
               jcc(EQ) t131(EFLAGS), $L9, $L6             #46
$L9: (1 ref)                                           #46
t133(EAX), EFLAGS = imul _i[EBP], 4                     #48
tv134-(EAX) = mov t133(EAX)                              #48
tv134-(EAX)<*27>, EFLAGS = add tv134-(EAX), &_amp;x         #48
t169(ECX)    = mov _j[EBP]                              #48
[tv134-(EAX)] = mov t169(ECX)                            #48

```

AppendixF.txt

```

t136(EAX), EFLAGS = imul _j[EBP], 4 #49
tv137-(EAX) = mov t136(EAX) #49
tv137-(EAX)<*29>, EFLAGS = add tv137-(EAX), _b[EBP] #49
[tv137-(EAX)] = mov 0 #49
tv139-(EAX) = mov _i[EBP] #50
tv139-(EAX)<*30>, EFLAGS = add tv139-(EAX), _j[EBP] #50
t140(EAX), EFLAGS = imul tv139-(EAX), 4 #50
tv141-(EAX) = mov t140(EAX) #50
tv141-(EAX)<*32>, EFLAGS = add tv141-(EAX), _a[EBP] #50
[tv141-(EAX)] = mov 0 #50
tv143-(EAX) = mov _i[EBP] #51
tv143-(EAX)<*33>, EFLAGS = sub tv143-(EAX), _j[EBP] #51
tv144-(EAX) = mov tv143-(EAX) #51
tv144-(EAX)<*34>, EFLAGS = add tv144-(EAX), 7 #51
t145(EAX), EFLAGS = imul tv144-(EAX), 4 #51
tv146-(EAX) = mov t145(EAX) #51
tv146-(EAX)<*36>, EFLAGS = add tv146-(EAX), &_c #51
[tv146-(EAX)] = mov 0 #51
t148(EFLAGS) = cmp(LT) _i[EBP], 8 #52
jcc(LT) t148(EFLAGS), $L11, $L10 #52
$L11: (1 ref) #52
tv149-(EAX) = mov 1 #54
tv149-(EAX)<*38>, EFLAGS = add tv149-(EAX), _i[EBP] #54
[ESP], {ESP} = push _b[EBP], {ESP} #54
[ESP], {ESP} = push _a[EBP], {ESP} #54
[ESP], {ESP} = push _q[EBP], {ESP} #54
[ESP], {ESP} = push tv149-(EAX), {ESP} #54
{-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7 FPUStatus} = call
try, $out[ESP], $out[ESP]+32, $out[ESP]+64, $out[ESP]+96, {-5}, {EAX ECX E
DX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7 FPUStatus} [Handler: $L1]
#54
jmp $L17 #65
$L17: (1 ref) #65
ESP, EFLAGS = add ESP, 16(0x00000010) #54
jmp $L16 #65
$L16: (1 ref) #65
t151(EAX) = mov _q[EBP] #55
t152(EFLAGS) = cmp(EQ) [t151(EAX)], 0 #55
jcc(EQ) t152(EFLAGS), $L13, $L12 #55
$L13: (1 ref) #55
t153(EAX), EFLAGS = imul _j[EBP], 4 #57
tv154-(EAX) = mov t153(EAX) #57
tv154-(EAX)<*42>, EFLAGS = add tv154-(EAX), _b[EBP] #57
[tv154-(EAX)] = mov 1 #57
tv156-(EAX) = mov _i[EBP] #58
tv156-(EAX)<*43>, EFLAGS = add tv156-(EAX), _j[EBP] #58
t157(EAX), EFLAGS = imul tv156-(EAX), 4 #58
tv158-(EAX) = mov t157(EAX) #58
tv158-(EAX)<*45>, EFLAGS = add tv158-(EAX), _a[EBP] #58
[tv158-(EAX)] = mov 1 #58
tv160-(EAX) = mov _i[EBP] #59
tv160-(EAX)<*46>, EFLAGS = sub tv160-(EAX), _j[EBP] #59
tv161-(EAX) = mov tv160-(EAX) #59

```

```

AppendixF.txt
tv161-(EAX)<*47>, EFLAGS = add tv161-(EAX), 7 #59
t162(EAX), EFLAGS = imul tv161-(EAX), 4 #59
tv163-(EAX) = mov t162(EAX) #59
tv163-(EAX)<*49>, EFLAGS = add tv163-(EAX), &_c #59
[tv163-(EAX)] = mov 1 #59
jmp $L12 #55
$L12: (2 refs) #55
jmp $L14 #62
$L10: (1 ref) #52
t165(EAX) = mov _q[EBP] #62
[t165(EAX)] = mov 1 #62
jmp $L14 #62
$L14: (2 refs) #62
jmp $L6 #46
$L6: (4 refs) #46
jmp $L2 #64
$L3: (2 refs) #42
jmp $L15 #65
$L1: (1 ref) #37
UNWIND _try #37
$L15: (1 ref) #65
EPILOGSTART #65
ESP = mov EBP #65
EBP, {ESP} = pop [ESP], {ESP} #65
{ESP} = ret {ESP} #65
EXITFUNC _try #65
END _try, {-7} #65

```

IR after Block Layout (flag Block Layout)

```

{-7} = START _try #37
_i<*4>, _q<*3>, _a<*2>, _b<*1> = ENTERFUNC _try #37
[ESP], {ESP} = push EBP, {ESP} #37
EBP = mov ESP #37
[ESP], {ESP} = push ESP, {ESP} #37
PROLOGEND #37
_j[EBP] = mov 0 #40
t110(EAX) = mov _q[EBP] #41
[t110(EAX)] = mov 0 #41
jmp $L2 #42
$L2: (2 refs) #42
t111(EAX) = mov _q[EBP] #42
t112(EFLAGS) = cmp(EQ) [t111(EAX)], 0 #42
jcc(EQ) t112(EFLAGS), $L4 #42
jmp $L3 #42
$L4: (1 ref) #42
t113(EFLAGS) = cmp(NE) _j[EBP], 8 #42
jcc(NE) t113(EFLAGS), $L5 #42
jmp $L3 #42
$L5: (1 ref) #42
tv114-(EAX) = mov 1 #44
tv114-(EAX)<*12>, EFLAGS = add tv114-(EAX), _j[EBP] #44

```

AppendixF.txt

```

_j[EBP]      = mov tv114-(EAX)      #44
t115(EAX)    = mov _q[EBP]         #45
[t115(EAX)]  = mov 0                #45
t116(EAX), EFLAGS = imul _j[EBP], 4 #46
tv117-(EAX)  = mov t116(EAX)       #46
tv117-(EAX)<*15>, EFLAGS = add tv117-(EAX), _b[EBP] #46
t119(EFLAGS) = cmp(EQ) [tv117-(EAX)], 1 #46
              jcc(EQ) t119(EFLAGS), $L7 #46
              jmp $L6                #46
$L7: (1 ref)                        #46
      tv120-(EAX) = mov _i[EBP]     #46
      tv120-(EAX)<*17>, EFLAGS = add tv120-(EAX), _j[EBP] #46
      t121(EAX), EFLAGS = imul tv120-(EAX), 4 #46
      tv122-(EAX) = mov t121(EAX)   #46
      tv122-(EAX)<*19>, EFLAGS = add tv122-(EAX), _a[EBP] #46
      t124(EFLAGS) = cmp(EQ) [tv122-(EAX)], 1 #46
                  jcc(EQ) t124(EFLAGS), $L8 #46
                  jmp $L6            #46
$L8: (1 ref)                        #46
      tv126-(EAX) = mov _i[EBP]     #46
      tv126-(EAX)<*21>, EFLAGS = sub tv126-(EAX), _j[EBP] #46
      tv127-(EAX) = mov tv126-(EAX) #46
      tv127-(EAX)<*22>, EFLAGS = add tv127-(EAX), 7 #46
      t128(EAX), EFLAGS = imul tv127-(EAX), 4 #46
      tv129-(EAX) = mov t128(EAX)   #46
      tv129-(EAX)<*24>, EFLAGS = add tv129-(EAX), &_amp;c #46
      t131(EFLAGS) = cmp(EQ) [tv129-(EAX)], 1 #46
                  jcc(EQ) t131(EFLAGS), $L9 #46
                  jmp $L6            #46
$L9: (1 ref)                        #46
      t133(EAX), EFLAGS = imul _i[EBP], 4 #48
      tv134-(EAX) = mov t133(EAX)   #48
      tv134-(EAX)<*27>, EFLAGS = add tv134-(EAX), &_amp;x #48
      t169(ECX) = mov _j[EBP]       #48
      [tv134-(EAX)] = mov t169(ECX) #48
      t136(EAX), EFLAGS = imul _j[EBP], 4 #49
      tv137-(EAX) = mov t136(EAX)   #49
      tv137-(EAX)<*29>, EFLAGS = add tv137-(EAX), _b[EBP] #49
      [tv137-(EAX)] = mov 0         #49
      tv139-(EAX) = mov _i[EBP]     #50
      tv139-(EAX)<*30>, EFLAGS = add tv139-(EAX), _j[EBP] #50
      t140(EAX), EFLAGS = imul tv139-(EAX), 4 #50
      tv141-(EAX) = mov t140(EAX)   #50
      tv141-(EAX)<*32>, EFLAGS = add tv141-(EAX), _a[EBP] #50
      [tv141-(EAX)] = mov 0         #50
      tv143-(EAX) = mov _i[EBP]     #51
      tv143-(EAX)<*33>, EFLAGS = sub tv143-(EAX), _j[EBP] #51
      tv144-(EAX) = mov tv143-(EAX) #51
      tv144-(EAX)<*34>, EFLAGS = add tv144-(EAX), 7 #51
      t145(EAX), EFLAGS = imul tv144-(EAX), 4 #51
      tv146-(EAX) = mov t145(EAX)   #51
      tv146-(EAX)<*36>, EFLAGS = add tv146-(EAX), &_amp;c #51
      [tv146-(EAX)] = mov 0         #51

```


AppendixF.txt

```

t148(EFLAGS) = cmp(LT) _i[EBP], 8 #52
                jcc(LT) t148(EFLAGS), $L11 #52
                jmp $L10 #52
$L11: (1 ref) #52
    tv149-(EAX) = mov 1 #54
    tv149-(EAX)<*38>, EFLAGS = add tv149-(EAX), _i[EBP] #54
    [ESP], {ESP} = push _b[EBP], {ESP} #54
    [ESP], {ESP} = push _a[EBP], {ESP} #54
    [ESP], {ESP} = push _q[EBP], {ESP} #54
    [ESP], {ESP} = push tv149-(EAX), {ESP} #54
    {-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7 FPUStatus} = call
    _try, $out[ESP], $out[ESP]+32, $out[ESP]+64, $out[ESP]+96, {-5}, {EAX ECX E
    DX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7 FPUStatus} [Handler: $L1]
    #54
                jmp $L17 #65
$L17: (1 ref) #65
    ESP, EFLAGS = add ESP, 16(0x00000010) #54
                jmp $L16 #65
$L16: (1 ref) #65
    t151(EAX) = mov _q[EBP] #55
    t152(EFLAGS) = cmp(EQ) [t151(EAX)], 0 #55
                jcc(EQ) t152(EFLAGS), $L13 #55
                jmp $L12 #55
$L13: (1 ref) #55
    t153(EAX), EFLAGS = imul _j[EBP], 4 #57
    tv154-(EAX) = mov t153(EAX) #57
    tv154-(EAX)<*42>, EFLAGS = add tv154-(EAX), _b[EBP] #57
    [tv154-(EAX)] = mov 1 #57
    tv156-(EAX) = mov _i[EBP] #58
    tv156-(EAX)<*43>, EFLAGS = add tv156-(EAX), _j[EBP] #58
    t157(EAX), EFLAGS = imul tv156-(EAX), 4 #58
    tv158-(EAX) = mov t157(EAX) #58
    tv158-(EAX)<*45>, EFLAGS = add tv158-(EAX), _a[EBP] #58
    [tv158-(EAX)] = mov 1 #58
    tv160-(EAX) = mov _i[EBP] #59
    tv160-(EAX)<*46>, EFLAGS = sub tv160-(EAX), _j[EBP] #59
    tv161-(EAX) = mov tv160-(EAX) #59
    tv161-(EAX)<*47>, EFLAGS = add tv161-(EAX), 7 #59
    t162(EAX), EFLAGS = imul tv161-(EAX), 4 #59
    tv163-(EAX) = mov t162(EAX) #59
    tv163-(EAX)<*49>, EFLAGS = add tv163-(EAX), &_amp;c #59
    [tv163-(EAX)] = mov 1 #59
                jmp $L12 #55
$L12: (2 refs) #55
                jmp $L14 #62
$L10: (1 ref) #52
    t165(EAX) = mov _q[EBP] #62
    [t165(EAX)] = mov 1 #62
                jmp $L14 #62
$L14: (2 refs) #62
                jmp $L6 #46
$L6: (4 refs) #46
                jmp $L2 #64

```

```

$!3:  (2 refs)                                     #42
                                           jmp $!15                                     #65
$!15:  (1 ref)                                     #65
                                           EPILOGSTART                                     #65
    ESP      = mov EBP                                     #65
    EBP, {ESP} = pop [ESP], {ESP}                         #65
    {ESP}     = ret {ESP}                                   #65
                                           EXITFUNC _try                                     #65
$!1:  (1 ref)                                     #37
                                           UNWIND _try                                     #37
                                           END _try, {-7}                                     #65

```

IR after Flow Optimization (flag FlowOpts)

```

{-7}      = START _try                                     #37
_i<*4>, _q<*3>, _a<*2>, _b<*1> = ENTERFUNC _try           #37
[ESP], {ESP} = push EBP, {ESP}                             #37
EBP        = mov ESP                                       #37
[ESP], {ESP} = push ESP, {ESP}                             #37
PROLOGEND                                           #37
_j[EBP]     = mov 0                                         #40
t110(EAX)    = mov _q[EBP]                                  #41
[t110(EAX)]  = mov 0                                         #41
$!2:  (6 refs)                                     #42
t111(EAX)    = mov _q[EBP]                                  #42
t112(EFLAGS) = cmp(EQ) [t111(EAX)], 0                       #42
               jcc(NE) t112(EFLAGS), $!15                   #42
t113(EFLAGS) = cmp(NE) _j[EBP], 8                           #42
               jcc(EQ) t113(EFLAGS), $!15                   #42
tv114-(EAX)  = mov 1                                         #44
tv114-(EAX)<*12>, EFLAGS = add tv114-(EAX), _j[EBP]         #44
_j[EBP]      = mov tv114-(EAX)                               #44
t115(EAX)    = mov _q[EBP]                                  #45
[t115(EAX)]  = mov 0                                         #45
t116(EAX), EFLAGS = imul _j[EBP], 4                         #46
tv117-(EAX)  = mov t116(EAX)                                #46
tv117-(EAX)<*15>, EFLAGS = add tv117-(EAX), _b[EBP]         #46
t119(EFLAGS) = cmp(EQ) [tv117-(EAX)], 1                     #46
               jcc(NE) t119(EFLAGS), $!2                   #46
tv120-(EAX)  = mov _i[EBP]                                  #46
tv120-(EAX)<*17>, EFLAGS = add tv120-(EAX), _j[EBP]         #46
t121(EAX), EFLAGS = imul tv120-(EAX), 4                     #46
tv122-(EAX)  = mov t121(EAX)                                #46
tv122-(EAX)<*19>, EFLAGS = add tv122-(EAX), _a[EBP]         #46
t124(EFLAGS) = cmp(EQ) [tv122-(EAX)], 1                     #46
               jcc(NE) t124(EFLAGS), $!2                   #46
tv126-(EAX)  = mov _i[EBP]                                  #46
tv126-(EAX)<*21>, EFLAGS = sub tv126-(EAX), _j[EBP]         #46
tv127-(EAX)  = mov tv126-(EAX)                              #46
tv127-(EAX)<*22>, EFLAGS = add tv127-(EAX), 7               #46
t128(EAX), EFLAGS = imul tv127-(EAX), 4                     #46
tv129-(EAX)  = mov t128(EAX)                               #46

```

```

AppendixF.txt
tv129-(EAX)<*24>, EFLAGS = add tv129-(EAX), &_c #46
t131(EFLAGS) = cmp(EQ) [tv129-(EAX)], 1 #46
jcc(NE) t131(EFLAGS), $L2 #46
t133(EAX), EFLAGS = imul _i[EBP], 4 #48
tv134-(EAX) = mov t133(EAX) #48
tv134-(EAX)<*27>, EFLAGS = add tv134-(EAX), &_x #48
t169(ECX) = mov _j[EBP] #48
[tv134-(EAX)] = mov t169(ECX) #48
t136(EAX), EFLAGS = imul _j[EBP], 4 #49
tv137-(EAX) = mov t136(EAX) #49
tv137-(EAX)<*29>, EFLAGS = add tv137-(EAX), _b[EBP] #49
[tv137-(EAX)] = mov 0 #49
tv139-(EAX) = mov _i[EBP] #50
tv139-(EAX)<*30>, EFLAGS = add tv139-(EAX), _j[EBP] #50
t140(EAX), EFLAGS = imul tv139-(EAX), 4 #50
tv141-(EAX) = mov t140(EAX) #50
tv141-(EAX)<*32>, EFLAGS = add tv141-(EAX), _a[EBP] #50
[tv141-(EAX)] = mov 0 #50
tv143-(EAX) = mov _i[EBP] #51
tv143-(EAX)<*33>, EFLAGS = sub tv143-(EAX), _j[EBP] #51
tv144-(EAX) = mov tv143-(EAX) #51
tv144-(EAX)<*34>, EFLAGS = add tv144-(EAX), 7 #51
t145(EAX), EFLAGS = imul tv144-(EAX), 4 #51
tv146-(EAX) = mov t145(EAX) #51
tv146-(EAX)<*36>, EFLAGS = add tv146-(EAX), &_c #51
[tv146-(EAX)] = mov 0 #51
t148(EFLAGS) = cmp(LT) _i[EBP], 8 #52
jcc(GE) t148(EFLAGS), $L10 #52
tv149-(EAX) = mov 1 #54
tv149-(EAX)<*38>, EFLAGS = add tv149-(EAX), _i[EBP] #54
[ESP], {ESP} = push _b[EBP], {ESP} #54
[ESP], {ESP} = push _a[EBP], {ESP} #54
[ESP], {ESP} = push _q[EBP], {ESP} #54
[ESP], {ESP} = push tv149-(EAX), {ESP} #54
{-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7 FPUStatus} = call
try, $out[ESP], $out[ESP]+32, $out[ESP]+64, $out[ESP]+96, {-5}, {EAX ECX E
DX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7 FPUStatus} [Handler: $L1]
#54
ESP, EFLAGS = add ESP, 16(0x00000010) #54
t151(EAX) = mov _q[EBP] #55
t152(EFLAGS) = cmp(EQ) [t151(EAX)], 0 #55
jcc(NE) t152(EFLAGS), $L2 #55
t153(EAX), EFLAGS = imul _j[EBP], 4 #57
tv154-(EAX) = mov t153(EAX) #57
tv154-(EAX)<*42>, EFLAGS = add tv154-(EAX), _b[EBP] #57
[tv154-(EAX)] = mov 1 #57
tv156-(EAX) = mov _i[EBP] #58
tv156-(EAX)<*43>, EFLAGS = add tv156-(EAX), _j[EBP] #58
t157(EAX), EFLAGS = imul tv156-(EAX), 4 #58
tv158-(EAX) = mov t157(EAX) #58
tv158-(EAX)<*45>, EFLAGS = add tv158-(EAX), _a[EBP] #58
[tv158-(EAX)] = mov 1 #58
tv160-(EAX) = mov _i[EBP] #59

```

AppendixF.txt

```

tv160-(EAX)<*46>, EFLAGS = sub tv160-(EAX), _j[EBP] #59
tv161-(EAX) = mov tv160-(EAX) #59
tv161-(EAX)<*47>, EFLAGS = add tv161-(EAX), 7 #59
t162(EAX), EFLAGS = imul tv161-(EAX), 4 #59
tv163-(EAX) = mov t162(EAX) #59
tv163-(EAX)<*49>, EFLAGS = add tv163-(EAX), &_c #59
[tv163-(EAX)] = mov 1 #59
        jmp $L2 #62
$L10: (1 ref) #52
        t165(EAX) = mov _q[EBP] #62
        [t165(EAX)] = mov 1 #62
        jmp $L2 #64
$L15: (2 refs) #65
        EPILOGSTART #65
        ESP = mov EBP #65
        EBP, {ESP} = pop [ESP], {ESP} #65
        {ESP} = ret {ESP} #65
        EXITFUNC _try #65
$L1: (1 ref) #37
        UNWIND _try #37
        END _try, {-7} #65

```

IR after Encoding, Listing, COFF Emission (flag Encode)

```

{-7} = START _try #37
_i<*4>, _q<*3>, _a<*2>, _b<*1> = ENTERFUNC _try #37
[ESP], {ESP} = push EBP, {ESP} #37
EBP = mov ESP #37
[ESP], {ESP} = push ESP, {ESP} #37
        PROLOGEND #37
_j[EBP] = mov 0 #40
t110(EAX) = mov _q[EBP] #41
[t110(EAX)] = mov 0 #41
$L2: (6 refs) #42
t111(EAX) = mov _q[EBP] #42
t112(EFLAGS) = cmp(EQ) [t111(EAX)], 0 #42
        jne(NE) t112(EFLAGS), $L15 #42
t113(EFLAGS) = cmp(NE) _j[EBP], 8 #42
        je(EQ) t113(EFLAGS), $L15 #42
tv114-(EAX) = mov 1 #44
tv114-(EAX)<*12>, EFLAGS = add tv114-(EAX), _j[EBP] #44
_j[EBP] = mov tv114-(EAX) #44
t115(EAX) = mov _q[EBP] #45
[t115(EAX)] = mov 0 #45
tv116-(EAX)<*14>, EFLAGS = imul _j[EBP], 4 #46
tv117-(EAX)<*15>, EFLAGS = add tv117-(EAX), _b[EBP] #46
t119(EFLAGS) = cmp(EQ) [tv117-(EAX)], 1 #46
        jne(NE) t119(EFLAGS), $L2 #46
tv120-(EAX) = mov _i[EBP] #46
tv120-(EAX)<*17>, EFLAGS = add tv120-(EAX), _j[EBP] #46
tv121-(EAX)<*18>, EFLAGS = imul tv120-(EAX), 4 #46
tv122-(EAX)<*19>, EFLAGS = add tv122-(EAX), _a[EBP] #46
t124(EFLAGS) = cmp(EQ) [tv122-(EAX)], 1 #46

```

AppendixF.txt

```

jne(NE) t124(EFLAGS), $L2 #46
tv126-(EAX) = mov _i[EBP] #46
tv126-(EAX)<*21>, EFLAGS = sub tv126-(EAX), _j[EBP] #46
tv127-(EAX)<*22>, EFLAGS = add tv127-(EAX), 7 #46
tv128-(EAX)<*23>, EFLAGS = imul tv127-(EAX), 4 #46
tv129-(EAX)<*24>, EFLAGS = add tv129-(EAX), &_c #46
t131(EFLAGS) = cmp(EQ) [tv129-(EAX)], 1 #46
jne(NE) t131(EFLAGS), $L2 #46
tv133-(EAX)<*26>, EFLAGS = imul _i[EBP], 4 #48
tv134-(EAX)<*27>, EFLAGS = add tv134-(EAX), &_x #48
t169(ECX) = mov _j[EBP] #48
[tv134-(EAX)] = mov t169(ECX) #48
tv136-(EAX)<*28>, EFLAGS = imul _j[EBP], 4 #49
tv137-(EAX)<*29>, EFLAGS = add tv137-(EAX), _b[EBP] #49
[tv137-(EAX)] = mov 0 #49
tv139-(EAX) = mov _i[EBP] #50
tv139-(EAX)<*30>, EFLAGS = add tv139-(EAX), _j[EBP] #50
tv140-(EAX)<*31>, EFLAGS = imul tv139-(EAX), 4 #50
tv141-(EAX)<*32>, EFLAGS = add tv141-(EAX), _a[EBP] #50
[tv141-(EAX)] = mov 0 #50
tv143-(EAX) = mov _i[EBP] #51
tv143-(EAX)<*33>, EFLAGS = sub tv143-(EAX), _j[EBP] #51
tv144-(EAX)<*34>, EFLAGS = add tv144-(EAX), 7 #51
tv145-(EAX)<*35>, EFLAGS = imul tv144-(EAX), 4 #51
tv146-(EAX)<*36>, EFLAGS = add tv146-(EAX), &_c #51
[tv146-(EAX)] = mov 0 #51
t148(EFLAGS) = cmp(LT) _i[EBP], 8 #52
jge(GE) t148(EFLAGS), $L10 #52
tv149-(EAX) = mov 1 #54
tv149-(EAX)<*38>, EFLAGS = add tv149-(EAX), _i[EBP] #54
[ESP], {ESP} = push _b[EBP], {ESP} #54
[ESP], {ESP} = push _a[EBP], {ESP} #54
[ESP], {ESP} = push _q[EBP], {ESP} #54
[ESP], {ESP} = push tv149-(EAX), {ESP} #54
{-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7 FPUSStatus} = call
try, $out[ESP], $out[ESP]+32, $out[ESP]+64, $out[ESP]+96, {-5}, {EAX ECX E
DX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7 FPUSStatus} [Handler: $L1]
#54
ESP, EFLAGS = add ESP, 16(0x00000010) #54
t151(EAX) = mov _q[EBP] #55
t152(EFLAGS) = cmp(EQ) [t151(EAX)], 0 #55
jne(NE) t152(EFLAGS), $L2 #55
tv153-(EAX)<*41>, EFLAGS = imul _j[EBP], 4 #57
tv154-(EAX)<*42>, EFLAGS = add tv154-(EAX), _b[EBP] #57
[tv154-(EAX)] = mov 1 #57
tv156-(EAX) = mov _i[EBP] #58
tv156-(EAX)<*43>, EFLAGS = add tv156-(EAX), _j[EBP] #58
tv157-(EAX)<*44>, EFLAGS = imul tv156-(EAX), 4 #58
tv158-(EAX)<*45>, EFLAGS = add tv158-(EAX), _a[EBP] #58
[tv158-(EAX)] = mov 1 #58
tv160-(EAX) = mov _i[EBP] #59
tv160-(EAX)<*46>, EFLAGS = sub tv160-(EAX), _j[EBP] #59
tv161-(EAX)<*47>, EFLAGS = add tv161-(EAX), 7 #59

```

```

AppendixF.txt
tv162-(EAX)<*48>, EFLAGS = imul tv161-(EAX), 4 #59
tv163-(EAX)<*49>, EFLAGS = add tv163-(EAX), &_c #59
[tv163-(EAX)] = mov 1 #59
                jmp $L2 #62
$L10: (1 ref) #52
    t165(EAX) = mov _q[EBP] #62
    [t165(EAX)] = mov 1 #62
                jmp $L2 #64
$L15: (2 refs) #65
    EPILOGSTART #65
    ESP = mov EBP #65
    EBP, {ESP} = pop [ESP], {ESP} #65
    {ESP} = ret {ESP} #65
    EXITFUNC _try #65
$L1: (1 ref) #37
    UNWIND _try #37
    END _try, {-7} #65

```

IR after LIR Phases (flag LIR Phases) [SubPhaseList]

```

{-7} = START _try #37
_i<*4>, _q<*3>, _a<*2>, _b<*1> = ENTERFUNC _try #37
[ESP], {ESP} = push EBP, {ESP} #37
EBP = mov ESP #37
[ESP], {ESP} = push ESP, {ESP} #37
    PROLOGEND #37
    _j[EBP] = mov 0 #40
    t110(EAX) = mov _q[EBP] #41
    [t110(EAX)] = mov 0 #41
$L2: (6 refs) #42
    t111(EAX) = mov _q[EBP] #42
    t112(EFLAGS) = cmp(EQ) [t111(EAX)], 0 #42
                    jne(NE) t112(EFLAGS), $L15 #42
    t113(EFLAGS) = cmp(NE) _j[EBP], 8 #42
                    je(EQ) t113(EFLAGS), $L15 #42
    tv114-(EAX) = mov 1 #44
    tv114-(EAX)<*12>, EFLAGS = add tv114-(EAX), _j[EBP] #44
    _j[EBP] = mov tv114-(EAX) #44
    t115(EAX) = mov _q[EBP] #45
    [t115(EAX)] = mov 0 #45
    tv116-(EAX)<*14>, EFLAGS = imul _j[EBP], 4 #46
    tv117-(EAX)<*15>, EFLAGS = add tv117-(EAX), _b[EBP] #46
    t119(EFLAGS) = cmp(EQ) [tv117-(EAX)], 1 #46
                    jne(NE) t119(EFLAGS), $L2 #46
    tv120-(EAX) = mov _i[EBP] #46
    tv120-(EAX)<*17>, EFLAGS = add tv120-(EAX), _j[EBP] #46
    tv121-(EAX)<*18>, EFLAGS = imul tv120-(EAX), 4 #46
    tv122-(EAX)<*19>, EFLAGS = add tv122-(EAX), _a[EBP] #46
    t124(EFLAGS) = cmp(EQ) [tv122-(EAX)], 1 #46
                    jne(NE) t124(EFLAGS), $L2 #46
    tv126-(EAX) = mov _i[EBP] #46
    tv126-(EAX)<*21>, EFLAGS = sub tv126-(EAX), _j[EBP] #46

```

AppendixF.txt

```

tv127-(EAX)<*22>, EFLAGS = add tv127-(EAX), 7 #46
tv128-(EAX)<*23>, EFLAGS = imul tv127-(EAX), 4 #46
tv129-(EAX)<*24>, EFLAGS = add tv129-(EAX), &_c #46
t131(EFLAGS) = cmp(EQ) [tv129-(EAX)], 1 #46
jne(NE) t131(EFLAGS), $L2 #46
tv133-(EAX)<*26>, EFLAGS = imul _i[EBP], 4 #48
tv134-(EAX)<*27>, EFLAGS = add tv134-(EAX), &_x #48
t169(ECX) = mov _j[EBP] #48
[tv134-(EAX)] = mov t169(ECX) #48
tv136-(EAX)<*28>, EFLAGS = imul _j[EBP], 4 #49
tv137-(EAX)<*29>, EFLAGS = add tv137-(EAX), _b[EBP] #49
[tv137-(EAX)] = mov 0 #49
tv139-(EAX) = mov _i[EBP] #50
tv139-(EAX)<*30>, EFLAGS = add tv139-(EAX), _j[EBP] #50
tv140-(EAX)<*31>, EFLAGS = imul tv139-(EAX), 4 #50
tv141-(EAX)<*32>, EFLAGS = add tv141-(EAX), _a[EBP] #50
[tv141-(EAX)] = mov 0 #50
tv143-(EAX) = mov _i[EBP] #51
tv143-(EAX)<*33>, EFLAGS = sub tv143-(EAX), _j[EBP] #51
tv144-(EAX)<*34>, EFLAGS = add tv144-(EAX), 7 #51
tv145-(EAX)<*35>, EFLAGS = imul tv144-(EAX), 4 #51
tv146-(EAX)<*36>, EFLAGS = add tv146-(EAX), &_c #51
[tv146-(EAX)] = mov 0 #51
t148(EFLAGS) = cmp(LT) _i[EBP], 8 #52
jge(GE) t148(EFLAGS), $L10 #52
tv149-(EAX) = mov 1 #54
tv149-(EAX)<*38>, EFLAGS = add tv149-(EAX), _i[EBP] #54
[ESP], {ESP} = push _b[EBP], {ESP} #54
[ESP], {ESP} = push _a[EBP], {ESP} #54
[ESP], {ESP} = push _q[EBP], {ESP} #54
[ESP], {ESP} = push tv149-(EAX), {ESP} #54
{-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7 FPUSStatus} = call
try, $out[ESP], $out[ESP]+32, $out[ESP]+64, $out[ESP]+96, {-5}, {EAX ECX E
DX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7 FPUSStatus} [Handler: $L1]
#54
ESP, EFLAGS = add ESP, 16(0x00000010) #54
t151(EAX) = mov _q[EBP] #55
t152(EFLAGS) = cmp(EQ) [t151(EAX)], 0 #55
jne(NE) t152(EFLAGS), $L2 #55
tv153-(EAX)<*41>, EFLAGS = imul _j[EBP], 4 #57
tv154-(EAX)<*42>, EFLAGS = add tv154-(EAX), _b[EBP] #57
[tv154-(EAX)] = mov 1 #57
tv156-(EAX) = mov _i[EBP] #58
tv156-(EAX)<*43>, EFLAGS = add tv156-(EAX), _j[EBP] #58
tv157-(EAX)<*44>, EFLAGS = imul tv156-(EAX), 4 #58
tv158-(EAX)<*45>, EFLAGS = add tv158-(EAX), _a[EBP] #58
[tv158-(EAX)] = mov 1 #58
tv160-(EAX) = mov _i[EBP] #59
tv160-(EAX)<*46>, EFLAGS = sub tv160-(EAX), _j[EBP] #59
tv161-(EAX)<*47>, EFLAGS = add tv161-(EAX), 7 #59
tv162-(EAX)<*48>, EFLAGS = imul tv161-(EAX), 4 #59
tv163-(EAX)<*49>, EFLAGS = add tv163-(EAX), &_c #59
[tv163-(EAX)] = mov 1 #59

```

```

                                jmp $L2                                #62
$L10: (1 ref)                                #52
    t165(EAX) = mov _q[EBP]                                #62
    [t165(EAX)] = mov 1                                #62
                                jmp $L2                                #64
$L15: (2 refs)                                #65
    EPILOGSTART                                #65
    ESP = mov EBP                                #65
    EBP, {ESP} = pop [ESP], {ESP}                                #65
    {ESP} = ret {ESP}                                #65
    EXITFUNC _try                                #65
$L1: (1 ref)                                #37
    UNWIND _try                                #37
    END _try, {-7}                                #65

/* IR after various phases for second part of program*/
IR after CIL Reader (flag cil)

    {-7} = START _main                                #68
    ENTERFUNC _main                                #68
    _doitagain = ASSIGN 1                                #78
    GOTO $L2                                #78
$L3: (1 ref)                                #78
    t109 = ASSIGN _doitagain                                #78
    t110 = ADD t109, 1                                #78
    _doitagain = ASSIGN t110                                #78
    GOTO $L2                                #78
$L2: (2 refs)                                #78
    t111 = CMP(LE) _doitagain, 6000(0x00001770)                                #78
    CBRANCH(LE) t111, $L5, $L4                                #78
$L5: (1 ref)                                #80
    _i = ASSIGN 0                                #81
    GOTO $L6                                #81
$L6: (2 refs)                                #81
    t112 = CMP(LE) _i, 16(0x00000010)                                #81
    CBRANCH(LE) t112, $L8, $L7                                #81
$L8: (1 ref)                                #83
    t113 = CMP(GE) _i, 1                                #83
    CBRANCH(GE) t113, $L10, $L9                                #83
$L10: (1 ref)                                #83
    t114 = CMP(LE) _i, 8                                #83
    CBRANCH(LE) t114, $L11, $L9                                #83
$L11: (1 ref)                                #83
    t116 = MUL _i, 4                                #83
    t117 = SUBSCRIPT &_a, t116                                #83
    t118 = ASSIGN t117                                #83
    [t118] = ASSIGN 1                                #83
    GOTO $L9                                #83
$L9: (3 refs)                                #84
    t119 = CMP(GE) _i, 2                                #84
    CBRANCH(GE) t119, $L13, $L12                                #84
$L13: (1 ref)                                #84

```


t121	= MUL _i, 4	#84
t122	= SUBSCRIPT &_b, t121	#84
t123	= ASSIGN t122	#84
[t123]	= ASSIGN 1	#84
	GOTO \$L12	#84
\$L12: (2 refs)		#84
t124	= CMP(LE) _i, 14(0x0000000e)	#85
	CBRANCH(LE) t124, \$L15, \$L14	#85
\$L15: (1 ref)		#85
t126	= MUL _i, 4	#85
t127	= SUBSCRIPT &_c, t126	#85
t128	= ASSIGN t127	#85
[t128]	= ASSIGN 1	#85
	GOTO \$L14	#85
\$L14: (2 refs)		#85
t129	= ADD _i, 1	#86
_i	= ASSIGN t129	#86
	GOTO \$L6	#87
\$L7: (1 ref)		#81
{-5}	= CALL _try, 1, &_q, &_b, &_a, {-5} [Handler: \$L1]	#89
t131	= CMP(EQ) _q, 1	#90
	CBRANCH(EQ) t131, \$L17, \$L16	#90
\$L17: (1 ref)		#90
_i	= ASSIGN 1	#91
	GOTO \$L18	#92
\$L18: (2 refs)		#92
t132	= CMP(LE) _i, 8	#92
	CBRANCH(LE) t132, \$L20, \$L19	#92
\$L20: (1 ref)		#92
t133	= ADD _i, 1	#93
_i	= ASSIGN t133	#93
	GOTO \$L18	#93
\$L19: (1 ref)		#92
	GOTO \$L21	#95
\$L16: (1 ref)		#90
{-5}	= CALL _printf, &\$SG1197, {-5} [Handler: \$L1]	#97
{-5}	= CALL _exit, 0, {-5} [Handler: \$L1]	#98
	GOTO \$L21	#95
\$L21: (2 refs)		#95
	GOTO \$L3	#100
\$L4: (1 ref)		#78
{-5}	= CALL _printf, &\$SG1198, {-5} [Handler: \$L1]	#101
	RETURN 0	#108
	GOTO \$L22	#108
\$L1: (4 refs)		#68
	UNWIND _main	#68
\$L22: (1 ref)		#108
	EXITFUNC _main	#109
	END _main, {-7}	#109

IR after Type Checker (flag Type Checker)

AppendixF.txt

{-7}	= START _main	#68
	ENTERFUNC _main	#68
_doitagain	= ASSIGN 1	#78
	GOTO \$L2	#78
\$L3: (1 ref)		#78
t109	= ASSIGN _doitagain	#78
t110	= ADD t109, 1	#78
_doitagain	= ASSIGN t110	#78
	GOTO \$L2	#78
\$L2: (2 refs)		#78
t111	= CMP(LE) _doitagain, 6000(0x00001770)	#78
	CBRANCH(LE) t111, \$L5, \$L4	#78
\$L5: (1 ref)		#78
_i	= ASSIGN 0	#80
	GOTO \$L6	#81
\$L6: (2 refs)		#81
t112	= CMP(LE) _i, 16(0x00000010)	#81
	CBRANCH(LE) t112, \$L8, \$L7	#81
\$L8: (1 ref)		#81
t113	= CMP(GE) _i, 1	#83
	CBRANCH(GE) t113, \$L10, \$L9	#83
\$L10: (1 ref)		#83
t114	= CMP(LE) _i, 8	#83
	CBRANCH(LE) t114, \$L11, \$L9	#83
\$L11: (1 ref)		#83
t116	= MUL _i, 4	#83
t117	= SUBSCRIPT &_a, t116	#83
t118	= ASSIGN t117	#83
[t118]	= ASSIGN 1	#83
	GOTO \$L9	#83
\$L9: (3 refs)		#83
t119	= CMP(GE) _i, 2	#84
	CBRANCH(GE) t119, \$L13, \$L12	#84
\$L13: (1 ref)		#84
t121	= MUL _i, 4	#84
t122	= SUBSCRIPT &_b, t121	#84
t123	= ASSIGN t122	#84
[t123]	= ASSIGN 1	#84
	GOTO \$L12	#84
\$L12: (2 refs)		#84
t124	= CMP(LE) _i, 14(0x0000000e)	#85
	CBRANCH(LE) t124, \$L15, \$L14	#85
\$L15: (1 ref)		#85
t126	= MUL _i, 4	#85
t127	= SUBSCRIPT &_c, t126	#85
t128	= ASSIGN t127	#85
[t128]	= ASSIGN 1	#85
	GOTO \$L14	#85
\$L14: (2 refs)		#85
t129	= ADD _i, 1	#86
_i	= ASSIGN t129	#86
	GOTO \$L6	#87
\$L7: (1 ref)		#81

AppendixF.txt

```

{-5}          = CALL _try, 1, &_q, &_b, &_a, {-5} [Handler: $L1]      #89
t131          = CMP(EQ) _q, 1                                         #90
               CBRANCH(EQ) t131, $L17, $L16                           #90
$L17: (1 ref)                                     #90
  _i          = ASSIGN 1                                              #91
               GOTO $L18                                              #92
$L18: (2 refs)                                     #92
  t132        = CMP(LE) _i, 8                                         #92
               CBRANCH(LE) t132, $L20, $L19                           #92
$L20: (1 ref)                                     #92
  t133        = ADD _i, 1                                             #93
  _i          = ASSIGN t133                                           #93
               GOTO $L18                                              #93
$L19: (1 ref)                                     #92
               GOTO $L21                                              #95
$L16: (1 ref)                                     #90
  {-5}        = CALL _printf, &$SG1197, {-5} [Handler: $L1]          #97
  {-5}        = CALL _exit, 0, {-5} [Handler: $L1]                   #98
               GOTO $L21                                              #95
$L21: (2 refs)                                     #95
               GOTO $L3                                               #100
$L4: (1 ref)                                     #78
  {-5}        = CALL _printf, &$SG1198, {-5} [Handler: $L1]          #101
               RETURN 0                                              #108
               GOTO $L22                                              #108
$L1: (4 refs)                                     #68
               UNWIND _main                                           #68
$L22: (1 ref)                                     #108
               EXITFUNC _main                                         #109
               END _main, {-7}                                        #109

```

IR after MIR Lower (flag MIRLower)

```

{-7}          = START _main                                           #68
               ENTERFUNC _main                                       #68
  _doitagain   = ASSIGN 1                                             #78
               GOTO $L2                                              #78
$L3: (1 ref)                                     #78
  t109         = ASSIGN _doitagain                                    #78
  t110         = ADD t109, 1                                          #78
  _doitagain   = ASSIGN t110                                          #78
               GOTO $L2                                              #78
$L2: (2 refs)                                     #78
  t111         = CMP(LE) _doitagain, 6000(0x00001770)                #78
               CBRANCH(LE) t111, $L5, $L4                            #78
$L5: (1 ref)                                     #80
  _i          = ASSIGN 0                                              #81
               GOTO $L6                                              #81
$L6: (2 refs)                                     #81
  t112         = CMP(LE) _i, 16(0x00000010)                          #81
               CBRANCH(LE) t112, $L8, $L7                            #81
$L8: (1 ref)                                     #81

```

t113	= CMP(GE) _i, 1	#83
	CBRANCH(GE) t113, \$L10, \$L9	#83
\$L10: (1 ref)		#83
t114	= CMP(LE) _i, 8	#83
	CBRANCH(LE) t114, \$L11, \$L9	#83
\$L11: (1 ref)		#83
t116	= MUL _i, 4	#83
t117	= ADD &_a, t116	#83
t118	= ASSIGN t117	#83
[t118]	= ASSIGN 1	#83
	GOTO \$L9	#83
\$L9: (3 refs)		#83
t119	= CMP(GE) _i, 2	#84
	CBRANCH(GE) t119, \$L13, \$L12	#84
\$L13: (1 ref)		#84
t121	= MUL _i, 4	#84
t122	= ADD &_b, t121	#84
t123	= ASSIGN t122	#84
[t123]	= ASSIGN 1	#84
	GOTO \$L12	#84
\$L12: (2 refs)		#84
t124	= CMP(LE) _i, 14(0x0000000e)	#85
	CBRANCH(LE) t124, \$L15, \$L14	#85
\$L15: (1 ref)		#85
t126	= MUL _i, 4	#85
t127	= ADD &_c, t126	#85
t128	= ASSIGN t127	#85
[t128]	= ASSIGN 1	#85
	GOTO \$L14	#85
\$L14: (2 refs)		#85
t129	= ADD _i, 1	#86
_i	= ASSIGN t129	#86
	GOTO \$L6	#87
\$L7: (1 ref)		#81
{-5}	= CALL _try, 1, &_q, &_b, &_a, {-5} [Handler: \$L1]	#89
t131	= CMP(EQ) _q, 1	#90
	CBRANCH(EQ) t131, \$L17, \$L16	#90
\$L17: (1 ref)		#90
_i	= ASSIGN 1	#91
	GOTO \$L18	#92
\$L18: (2 refs)		#92
t132	= CMP(LE) _i, 8	#92
	CBRANCH(LE) t132, \$L20, \$L19	#92
\$L20: (1 ref)		#92
t133	= ADD _i, 1	#93
_i	= ASSIGN t133	#93
	GOTO \$L18	#93
\$L19: (1 ref)		#92
	GOTO \$L21	#95
\$L16: (1 ref)		#90
{-5}	= CALL _printf, &\$SG1197, {-5} [Handler: \$L1]	#97
{-5}	= CALL _exit, 0, {-5} [Handler: \$L1]	#98
	GOTO \$L21	#95

AppendixF.txt

```

$121:  (2 refs)                                     #95
      GOTO $13                                       #100
$14:  (1 ref)                                     #78
      {-5} = CALL _printf, &$SG1198, {-5} [Handler: $11] #101
      RETURN 0                                       #108
      GOTO $122                                      #108
$11:  (4 refs)                                     #68
      UNWIND _main                                   #68
$122:  (1 ref)                                     #108
      EXITFUNC _main                                #109
      END _main, {-7}                               #109

```

IR after Ssa Construction and Optimization (flag Ssa)

```

==== Block 1 Pred() Succ(2) next 2 pre 1 post 58 iDom 1 df
      {-7}, {-1} = START _main                                     #68
==== Block 2 Pred(1) Succ(4) prev 1 next 4 pre 2 post 57 iDom 1 df
      ENTERFUNC _main                                             #68
      _doitagain<*1> = ASSIGN 1                                     #78
      GOTO $12                                                     #78
==== Block 4 Pred(3,2) Succ(25,5) prev 2 next 5 pre 3 post 56 iDom 2 df 4
$12:  (2 refs)                                                     #78
      _doitagain<*2> = PHI _doitagain<3>, _doitagain<1>           #109
      tv111-<*6> = CMP(LE) _doitagain<2>, 6000(0x00001770)        #78
      CBRANCH(LE) tv111-<6>, $15, $14                             #78
==== Block 5 Pred(4) Succ(6) prev 4 next 6 pre 14 post 55 iDom 4 df 4,27
$15:  (1 ref)                                                     #78
      _i<*7> = ASSIGN 0                                           #80
      GOTO $16                                                     #81
==== Block 6 Pred(14,5) Succ(15,7) prev 5 next 7 pre 15 post 54 iDom 5 df 4,
6,27
$16:  (2 refs)                                                     #81
      _i<*8> = PHI _i<12>, _i<7>                                   #109
      tv112-<*13> = CMP(LE) _i<8>, 16(0x00000010)                 #81
      CBRANCH(LE) tv112-<13>, $18, $17                             #81
==== Block 7 Pred(6) Succ(10,8) prev 6 next 8 pre 38 post 53 iDom 6 df 6
$18:  (1 ref)                                                     #81
      tv113-<*14> = CMP(GE) _i<8>, 1                               #83
      CBRANCH(GE) tv113-<14>, $110, $19                           #83
==== Block 8 Pred(7) Succ(10,9) prev 7 next 9 pre 49 post 52 iDom 7 df 10
$110: (1 ref)                                                     #83
      tv114-<*15> = CMP(LE) _i<8>, 8                               #83
      CBRANCH(LE) tv114-<15>, $111, $19                           #83
==== Block 9 Pred(8) Succ(10) prev 8 next 10 pre 50 post 51 iDom 8 df 10
$111: (1 ref)                                                     #83
      tv116-<*16> = MUL _i<8>, 4                                   #83
      tv117-<*17> = ADD &_a, tv116-<16>                           #83
      [tv117-<17>] = ASSIGN 1                                     #83
      GOTO $19                                                     #83
==== Block 10 Pred(9,8,7) Succ(12,11) prev 9 next 11 pre 39 post 48 iDom 7 d
f 6

```

AppendixF.txt

```

$L9: (3 refs) #83
    tv119-<*>18> = CMP(GE) _i<8>, 2 #84
                CBRANCH(GE) tv119-<18>, $L13, $L12 #84
==== Block 11 Pred(10) Succ(12) prev 10 next 12 pre 46 post 47 iDom 10 df 12
$L13: (1 ref) #84
    tv121-<*>19> = MUL _i<8>, 4 #84
    tv122-<*>20> = ADD &_b, tv121-<19> #84
    [tv122-<20>] = ASSIGN 1 #84
                GOTO $L12 #84
==== Block 12 Pred(11,10) Succ(14,13) prev 11 next 13 pre 40 post 45 iDom 10
df 6
$L12: (2 refs) #84
    tv124-<*>21> = CMP(LE) _i<8>, 14(0x0000000e) #85
                CBRANCH(LE) tv124-<21>, $L15, $L14 #85
==== Block 13 Pred(12) Succ(14) prev 12 next 14 pre 43 post 44 iDom 12 df 14
$L15: (1 ref) #85
    tv126-<*>22> = MUL _i<8>, 4 #85
    tv127-<*>23> = ADD &_c, tv126-<22> #85
    [tv127-<23>] = ASSIGN 1 #85
                GOTO $L14 #85
==== Block 14 Pred(13,12) Succ(6) prev 13 next 15 pre 41 post 42 iDom 12 df
6
$L14: (2 refs) #85
    tv129-<*>24> = ADD _i<8>, 1 #86
    _i<*>12> = ASSIGN tv129-<24> #86
                GOTO $L6 #87
==== Block 15 Pred(6) Succ(16,27) prev 14 next 16 pre 16 post 37 iDom 6 df 4
,27
$L7: (1 ref) #81
    {-5} = CALL _try, 1, &_q, &_b, &_a, {-5} [Handler: $L1] #89
                GOTO $L23 #109
==== Block 16 Pred(15) Succ(21,17) prev 15 next 17 pre 17 post 36 iDom 15 df
4,27
$L23: (1 ref) #109
    tv131-<*>25> = CMP(EQ) _q, 1 #90
                CBRANCH(EQ) tv131-<25>, $L17, $L16 #90
==== Block 17 Pred(16) Succ(18) prev 16 next 18 pre 28 post 35 iDom 16 df 24
$L17: (1 ref) #90
    _i<*>9> = ASSIGN 1 #91
                GOTO $L18 #92
==== Block 18 Pred(19,17) Succ(20,19) prev 17 next 19 pre 29 post 34 iDom 17
df 18,24
$L18: (2 refs) #92
    _i<*>10> = PHI _i<11>, _i<9> #109
    tv132-<*>26> = CMP(LE) _i<10>, 8 #92
                CBRANCH(LE) tv132-<26>, $L20, $L19 #92
==== Block 19 Pred(18) Succ(18) prev 18 next 20 pre 32 post 33 iDom 18 df 18
$L20: (1 ref) #92
    tv133-<*>27> = ADD _i<10>, 1 #93
    _i<*>11> = ASSIGN tv133-<27> #93
                GOTO $L18 #93
==== Block 20 Pred(18) Succ(24) prev 19 next 21 pre 30 post 31 iDom 18 df 24
$L19: (1 ref) #92

```

AppendixF.txt

```

GOTO $L21 #95
==== Block 21 Pred(16) Succ(22,27) prev 20 next 22 pre 18 post 27 iDom 16 df
24,27
$L16: (1 ref) #90
{-5} = CALL _printf, &$SG1197, {-5} [Handler: $L1] #97
GOTO $L24 #109
==== Block 22 Pred(21) Succ(23,27) prev 21 next 23 pre 19 post 26 iDom 21 df
24,27
$L24: (1 ref) #109
{-5} = CALL _exit, 0, {-5} [Handler: $L1] #98
GOTO $L25 #109
==== Block 23 Pred(22) Succ(24) prev 22 next 24 pre 20 post 25 iDom 22 df 24
$L25: (1 ref) #109
GOTO $L21 #95
==== Block 24 Pred(23,20) Succ(3) prev 23 next 3 pre 21 post 24 iDom 16 df 4
$L21: (2 refs) #95
GOTO $L3 #100
==== Block 3 Pred(24) Succ(4) prev 24 next 25 pre 22 post 23 iDom 24 df 4
$L3: (1 ref) #78
tv109-<*4> = ASSIGN _doitagain<2> #78
tv110-<*5> = ADD tv109-<4>, 1 #78
_doitagain<*3> = ASSIGN tv110-<5> #78
GOTO $L2 #78
==== Block 25 Pred(4) Succ(26,27) prev 3 next 27 pre 4 post 13 iDom 4 df 27,
29
$L4: (1 ref) #78
{-5} = CALL _printf, &$SG1198, {-5} [Handler: $L1] #101
GOTO $L26 #109
==== Block 27 Pred(25,22,21,15) Succ(29) prev 25 next 26 pre 11 post 12 iDom
4 df 29
$L1: (4 refs) #68
UNWIND _main #68
==== Block 26 Pred(25) Succ(28) prev 27 next 28 pre 5 post 10 iDom 25 df 29
$L26: (1 ref) #109
RETURN 0 #108
GOTO $L22 #108
==== Block 28 Pred(26) Succ(29) prev 26 next 29 pre 6 post 9 iDom 26 df 29
$L22: (1 ref) #108
EXITFUNC _main #109
==== Block 29 Pred(28,27) Succ() prev 28 pre 7 post 8 iDom 4 df
END _main, {-7} #109

```

IR after Ssa Info Destruction (flag Ssa)

```

{-7} = START _main #68
ENTERFUNC _main #68
_doitagain<*1> = ASSIGN 1 #78
GOTO $L2 #78
$L3: (1 ref) #78
t109 = ASSIGN _doitagain #78
t110 = ADD t109, 1 #78
_doitagain<*3> = ASSIGN t110 #78

```

AppendixF.txt

```

GOTO $L2 #78
$L2: (2 refs) #78
    t111 = CMP(LE) _doitagain, 6000(0x00001770) #78
        CBRANCH(LE) t111, $L5, $L4 #78
$L5: (1 ref) #78
    _i<*7> = ASSIGN 0 #80
        GOTO $L6 #81
$L6: (2 refs) #81
    t112 = CMP(LE) _i, 16(0x00000010) #81
        CBRANCH(LE) t112, $L8, $L7 #81
$L8: (1 ref) #81
    t113 = CMP(GE) _i, 1 #83
        CBRANCH(GE) t113, $L10, $L9 #83
$L10: (1 ref) #83
    t114 = CMP(LE) _i, 8 #83
        CBRANCH(LE) t114, $L11, $L9 #83
$L11: (1 ref) #83
    t116 = MUL _i, 4 #83
    t117 = ADD &_a, t116 #83
    [t117] = ASSIGN 1 #83
        GOTO $L9 #83
$L9: (3 refs) #83
    t119 = CMP(GE) _i, 2 #84
        CBRANCH(GE) t119, $L13, $L12 #84
$L13: (1 ref) #84
    t121 = MUL _i, 4 #84
    t122 = ADD &_b, t121 #84
    [t122] = ASSIGN 1 #84
        GOTO $L12 #84
$L12: (2 refs) #84
    t124 = CMP(LE) _i, 14(0x0000000e) #85
        CBRANCH(LE) t124, $L15, $L14 #85
$L15: (1 ref) #85
    t126 = MUL _i, 4 #85
    t127 = ADD &_c, t126 #85
    [t127] = ASSIGN 1 #85
        GOTO $L14 #85
$L14: (2 refs) #85
    t129 = ADD _i, 1 #86
    _i<*12> = ASSIGN t129 #86
        GOTO $L6 #87
$L7: (1 ref) #81
    {-5} = CALL _try, 1, &_q, &_b, &_a, {-5} [Handler: $L1] #89
        GOTO $L23 #109
$L23: (1 ref) #109
    t131 = CMP(EQ) _q, 1 #90
        CBRANCH(EQ) t131, $L17, $L16 #90
$L17: (1 ref) #90
    _i<*9> = ASSIGN 1 #91
        GOTO $L18 #92
$L18: (2 refs) #92
    t132 = CMP(LE) _i, 8 #92
        CBRANCH(LE) t132, $L20, $L19 #92

```


AppendixF.txt

\$L20: (1 ref)		#92
t133	= ADD _i, 1	#93
_i<*11>	= ASSIGN t133	#93
	GOTO \$L18	#93
\$L19: (1 ref)		#92
	GOTO \$L21	#95
\$L16: (1 ref)		#90
{-5}	= CALL _printf, &\$SG1197, {-5} [Handler: \$L1]	#97
	GOTO \$L24	#109
\$L24: (1 ref)		#109
{-5}	= CALL _exit, 0, {-5} [Handler: \$L1]	#98
	GOTO \$L25	#109
\$L25: (1 ref)		#109
	GOTO \$L21	#95
\$L21: (2 refs)		#95
	GOTO \$L3	#100
\$L4: (1 ref)		#78
{-5}	= CALL _printf, &\$SG1198, {-5} [Handler: \$L1]	#101
	GOTO \$L26	#109
\$L26: (1 ref)		#109
	RETURN 0	#108
	GOTO \$L22	#108
\$L1: (4 refs)		#68
	UNWIND _main	#68
\$L22: (1 ref)		#108
	EXITFUNC _main	#109
	END _main, {-7}	#109

IR after Lower (flag Lower)

{-7}	= START _main	#68
	ENTERFUNC _main	#68
	PROLOGEND	#68
_doitagain[_FP]	= mov 1	#78
	jmp \$L2	#78
\$L3: (1 ref)		#78
t109(_rd)	= mov _doitagain[_FP]	#78
tv110-(_rd)	= mov t109(_rd)	#78
tv110-(_rd)<*5>	, EFLAGS = add tv110-(_rd), 1	#78
_doitagain[_FP]	= mov tv110-(_rd)	#78
	jmp \$L2	#78
\$L2: (2 refs)		#78
t111(EFLAGS)	= cmp(LE) _doitagain[_FP], 6000(0x00001770)	#78
	jcc(LE) t111(EFLAGS), \$L5, \$L4	#78
\$L5: (1 ref)		#78
_i[_FP]	= mov 0	#80
	jmp \$L6	#81
\$L6: (2 refs)		#81
t112(EFLAGS)	= cmp(LE) _i[_FP], 16(0x00000010)	#81
	jcc(LE) t112(EFLAGS), \$L8, \$L7	#81
\$L8: (1 ref)		#81
t113(EFLAGS)	= cmp(GE) _i[_FP], 1	#83

AppendixF.txt

```

                                jcc(GE) t113(EFLAGS), $L10, $L9          #83
$L10: (1 ref)                  #83
                                t114(EFLAGS) = cmp(LE) _i[_FP], 8      #83
                                jcc(LE) t114(EFLAGS), $L11, $L9        #83
$L11: (1 ref)                  #83
                                t116(_rd), EFLAGS = imul _i[_FP], 4    #83
                                tv117-(_rd) = mov t116(_rd)           #83
                                tv117-(_rd)<*17>, EFLAGS = add tv117-(_rd), &_a #83
                                [tv117-(_rd)] = mov 1                  #83
                                jmp $L9                                #83
$L9: (3 refs)                  #83
                                t119(EFLAGS) = cmp(GE) _i[_FP], 2      #84
                                jcc(GE) t119(EFLAGS), $L13, $L12        #84
$L13: (1 ref)                  #84
                                t121(_rd), EFLAGS = imul _i[_FP], 4    #84
                                tv122-(_rd) = mov t121(_rd)           #84
                                tv122-(_rd)<*20>, EFLAGS = add tv122-(_rd), &_b #84
                                [tv122-(_rd)] = mov 1                  #84
                                jmp $L12                                #84
$L12: (2 refs)                  #84
                                t124(EFLAGS) = cmp(LE) _i[_FP], 14(0x0000000e) #85
                                jcc(LE) t124(EFLAGS), $L15, $L14        #85
$L15: (1 ref)                  #85
                                t126(_rd), EFLAGS = imul _i[_FP], 4    #85
                                tv127-(_rd) = mov t126(_rd)           #85
                                tv127-(_rd)<*23>, EFLAGS = add tv127-(_rd), &_c #85
                                [tv127-(_rd)] = mov 1                  #85
                                jmp $L14                                #85
$L14: (2 refs)                  #85
                                tv129-(_rd) = mov 1                    #86
                                tv129-(_rd)<*24>, EFLAGS = add tv129-(_rd), _i[_FP] #86
                                _i[_FP] = mov tv129-(_rd)              #86
                                jmp $L6                                #87
$L7: (1 ref)                   #81
                                [ESP], {ESP} = push &_a, {ESP}         #89
                                [ESP], {ESP} = push &_b, {ESP}         #89
                                t140(_rd) = lea &_q[_FP]               #89
                                [ESP], {ESP} = push t140(_rd), {ESP}    #89
                                [ESP], {ESP} = push 1, {ESP}            #89
                                {-5}, {EAX ECX ESP EFLAGS} = call _try, $out[ESP], $out[ESP]+32, $out[ESP]
                                ]+64, $out[ESP]+96, {-5}, {EAX ECX ESP EFLAGS} [Handler: $L1]
#89
                                ESP, EFLAGS = add ESP, 16(0x00000010)    #89
                                jmp $L23                                #109
$L23: (1 ref)                  #109
                                t131(EFLAGS) = cmp(EQ) _q[_FP], 1      #90
                                jcc(EQ) t131(EFLAGS), $L17, $L16        #90
$L17: (1 ref)                  #90
                                _i[_FP] = mov 1                        #91
                                jmp $L18                                #92
$L18: (2 refs)                  #92
                                t132(EFLAGS) = cmp(LE) _i[_FP], 8      #92
                                jcc(LE) t132(EFLAGS), $L20, $L19        #92

```

AppendixF.txt

```

$L20: (1 ref)                                     #92
    tv133-(_rd) = mov 1                             #93
    tv133-(_rd)<*27>, EFLAGS = add tv133-(_rd), _i[_FP] #93
    _i[_FP] = mov tv133-(_rd)                       #93
    jmp $L18                                         #93
$L19: (1 ref)                                     #92
    jmp $L21                                         #95
$L16: (1 ref)                                     #90
    [ESP], {ESP} = push &$SG1197, {ESP}             #97
    {-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7 FPUStatus} = call
    _printf, $out[ESP], {-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7
    FPUStatus} [Handler: $L1]
    #97
    ESP, EFLAGS = add ESP, 4                         #97
    jmp $L24                                         #109
$L24: (1 ref)                                     #109
    [ESP], {ESP} = push 0, {ESP}                     #98
    {-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7 FPUStatus} = call
    _exit, $out[ESP], {-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7 F
    PUStatus} [Handler: $L1]
    #98
    ESP, EFLAGS = add ESP, 4                         #98
    jmp $L25                                         #109
$L25: (1 ref)                                     #109
    jmp $L21                                         #95
$L21: (2 refs)                                    #95
    jmp $L3                                          #100
$L4: (1 ref)                                       #78
    [ESP], {ESP} = push &$SG1198, {ESP}             #101
    {-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7 FPUStatus} = call
    _printf, $out[ESP], {-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7
    FPUStatus} [Handler: $L1]
    #101
    ESP, EFLAGS = add ESP, 4                         #101
    jmp $L26                                         #109
$L26: (1 ref)                                     #109
    t142(EAX) = mov 0                                #108
    jmp $L22                                         #108
$L1: (4 refs)                                     #68
    UNWIND _main                                     #68
$L22: (1 ref)                                     #108
    EPILOGSTART                                     #109
    EXITFUNC _main, t142(EAX)                       #109
    END _main, {-7}                                 #109

```

IR after Linear Scan Register Allocation (flag LinearScan)

```

{-7} = START _main                                #68
    ENTERFUNC _main                                #68
    PROLOGEND                                       #68
    _doitagain[_FP] = mov 1                        #78
    jmp $L2                                         #78

```

AppendixF.txt

```

$!3:  (1 ref)                                     #78
    t109(EAX) = mov _doitagain[_FP]                #78
    tv110-(EAX) = mov t109(EAX)                    #78
    tv110-(EAX)<*5>, EFLAGS = add tv110-(EAX), 1    #78
    _doitagain[_FP] = mov tv110-(EAX)              #78
    jmp $!2                                         #78
$!2:  (2 refs)                                     #78
    t111(EFLAGS) = cmp(LE) _doitagain[_FP], 6000(0x00001770) #78
    jcc(LE) t111(EFLAGS), $!5, $!4                 #78
$!5:  (1 ref)                                     #78
    _i[_FP] = mov 0                                #80
    jmp $!6                                         #81
$!6:  (2 refs)                                     #81
    t112(EFLAGS) = cmp(LE) _i[_FP], 16(0x00000010)  #81
    jcc(LE) t112(EFLAGS), $!8, $!7                 #81
$!8:  (1 ref)                                     #81
    t113(EFLAGS) = cmp(GE) _i[_FP], 1              #83
    jcc(GE) t113(EFLAGS), $!10, $!9                #83
$!10: (1 ref)                                     #83
    t114(EFLAGS) = cmp(LE) _i[_FP], 8              #83
    jcc(LE) t114(EFLAGS), $!11, $!9                #83
$!11: (1 ref)                                     #83
    t116(EAX), EFLAGS = imul _i[_FP], 4            #83
    tv117-(EAX) = mov t116(EAX)                    #83
    tv117-(EAX)<*17>, EFLAGS = add tv117-(EAX), &_a #83
    [tv117-(EAX)] = mov 1                          #83
    jmp $!9                                         #83
$!9:  (3 refs)                                     #83
    t119(EFLAGS) = cmp(GE) _i[_FP], 2              #84
    jcc(GE) t119(EFLAGS), $!13, $!12               #84
$!13: (1 ref)                                     #84
    t121(EAX), EFLAGS = imul _i[_FP], 4            #84
    tv122-(EAX) = mov t121(EAX)                    #84
    tv122-(EAX)<*20>, EFLAGS = add tv122-(EAX), &_b #84
    [tv122-(EAX)] = mov 1                          #84
    jmp $!12                                        #84
$!12: (2 refs)                                     #84
    t124(EFLAGS) = cmp(LE) _i[_FP], 14(0x0000000e) #85
    jcc(LE) t124(EFLAGS), $!15, $!14               #85
$!15: (1 ref)                                     #85
    t126(EAX), EFLAGS = imul _i[_FP], 4            #85
    tv127-(EAX) = mov t126(EAX)                    #85
    tv127-(EAX)<*23>, EFLAGS = add tv127-(EAX), &_c #85
    [tv127-(EAX)] = mov 1                          #85
    jmp $!14                                        #85
$!14: (2 refs)                                     #85
    tv129-(EAX) = mov 1                            #86
    tv129-(EAX)<*24>, EFLAGS = add tv129-(EAX), _i[_FP] #86
    _i[_FP] = mov tv129-(EAX)                      #86
    jmp $!6                                         #87
$!7:  (1 ref)                                     #81
    [ESP], {ESP} = push &_a, {ESP}                 #89
    [ESP], {ESP} = push &_b, {ESP}                 #89

```

AppendixF.txt

```

t140(EAX)    = lea &_q[_FP]                                #89
[ESP], {ESP} = push t140(EAX), {ESP}                       #89
[ESP], {ESP} = push 1, {ESP}                               #89
{-5}, {EAX ECX ESP EFLAGS} = call _try, $out[ESP], $out[ESP]+32, $out[ESP
]+64, $out[ESP]+96, {-5}, {EAX ECX ESP EFLAGS} [Handler: $L1]
#89
        jmp $L27                                           #109
$L27:  (1 ref)                                           #109
        ESP, EFLAGS = add ESP, 16(0x00000010)              #89
        jmp $L23                                           #109
$L23:  (1 ref)                                           #109
        t131(EFLAGS) = cmp(EQ) _q[_FP], 1                 #90
        jcc(EQ) t131(EFLAGS), $L17, $L16                  #90
$L17:  (1 ref)                                           #90
        _i[_FP]      = mov 1                               #91
        jmp $L18                                           #92
$L18:  (2 refs)                                           #92
        t132(EFLAGS) = cmp(LE) _i[_FP], 8                 #92
        jcc(LE) t132(EFLAGS), $L20, $L19                  #92
$L20:  (1 ref)                                           #92
        tv133-(EAX) = mov 1                               #93
        tv133-(EAX)<*27>, EFLAGS = add tv133-(EAX), _i[_FP] #93
        _i[_FP]      = mov tv133-(EAX)                   #93
        jmp $L18                                           #93
$L19:  (1 ref)                                           #92
        jmp $L21                                           #95
$L16:  (1 ref)                                           #90
        [ESP], {ESP} = push &$SG1197, {ESP}                #97
        {-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7 FPUStatus} = call
_printf, $out[ESP], {-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7 F
PUStatus} [Handler: $L1]
        #97
        jmp $L28                                           #109
$L28:  (1 ref)                                           #109
        ESP, EFLAGS = add ESP, 4                           #97
        jmp $L24                                           #109
$L24:  (1 ref)                                           #109
        [ESP], {ESP} = push 0, {ESP}                       #98
        {-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7 FPUStatus} = call
_exit, $out[ESP], {-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7 F
PUStatus} [Handler: $L1]
        #98
        jmp $L29                                           #109
$L29:  (1 ref)                                           #109
        ESP, EFLAGS = add ESP, 4                           #98
        jmp $L25                                           #109
$L25:  (1 ref)                                           #109
        jmp $L21                                           #95
$L21:  (2 refs)                                           #95
        jmp $L3                                             #100
$L4:   (1 ref)                                           #78
        [ESP], {ESP} = push &$SG1198, {ESP}                #101
        {-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7 FPUStatus} = call

```

AppendixF.txt

```

_printf, $out[ESP], {-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7
FPUStatus} [Handler: $L1]
    #101
        jmp $L30
L30:  (1 ref)
    ESP, EFLAGS = add ESP, 4
        jmp $L26
L26:  (1 ref)
    t142(EAX) = mov 0
        jmp $L22
L1:   (4 refs)
    UNWIND _main
L22:  (1 ref)
    EPILOGSTART
    EXITFUNC _main, t142(EAX)
    END main, {-7}

```

IR after Stack Allocation (flag StackAlloc)

```

{-7}          = START _main                                #68
               ENTERFUNC _main                             #68
               PROLOGEND                                    #68
   _doitagain[_FP] = mov 1                                  #78
               jmp $L2                                      #78
$L3:  (1 ref)                                           #78
   t109(EAX)    = mov _doitagain[_FP]                     #78
   tv110-(EAX) = mov t109(EAX)                            #78
   tv110-(EAX)<*5>, EFLAGS = add tv110-(EAX), 1           #78
   _doitagain[_FP] = mov tv110-(EAX)                     #78
               jmp $L2                                      #78
$L2:  (2 refs)                                           #78
   t111(EFLAGS) = cmp(LE) _doitagain[_FP], 6000(0x00001770) #78
               jcc(LE) t111(EFLAGS), $L5, $L4             #78
$L5:  (1 ref)                                           #78
   _i[_FP]      = mov 0                                    #80
               jmp $L6                                      #81
$L6:  (2 refs)                                           #81
   t112(EFLAGS) = cmp(LE) _i[_FP], 16(0x00000010)         #81
               jcc(LE) t112(EFLAGS), $L8, $L7             #81
$L8:  (1 ref)                                           #81
   t113(EFLAGS) = cmp(GE) _i[_FP], 1                     #83
               jcc(GE) t113(EFLAGS), $L10, $L9            #83
$L10: (1 ref)                                           #83
   t114(EFLAGS) = cmp(LE) _i[_FP], 8                     #83
               jcc(LE) t114(EFLAGS), $L11, $L9            #83
$L11: (1 ref)                                           #83
   t116(EAX), EFLAGS = imul _i[_FP], 4                   #83
   tv117-(EAX) = mov t116(EAX)                            #83
   tv117-(EAX)<*17>, EFLAGS = add tv117-(EAX), &_a       #83
   [tv117-(EAX)] = mov 1                                  #83
               jmp $L9                                      #83
$L9:  (3 refs)                                           #83

```

AppendixF.txt

```

t119(EFLAGS) = cmp(GE) _i[_FP], 2                                #84
                jcc(GE) t119(EFLAGS), $L13, $L12                #84
$L13: (1 ref)                                                    #84
    t121(EAX), EFLAGS = imul _i[_FP], 4                          #84
    tv122-(EAX) = mov t121(EAX)                                  #84
    tv122-(EAX)<*20>, EFLAGS = add tv122-(EAX), &_amp;b           #84
    [tv122-(EAX)] = mov 1                                        #84
                jmp $L12                                         #84
$L12: (2 refs)                                                    #84
    t124(EFLAGS) = cmp(LE) _i[_FP], 14(0x0000000e)              #85
                jcc(LE) t124(EFLAGS), $L15, $L14                #85
$L15: (1 ref)                                                    #85
    t126(EAX), EFLAGS = imul _i[_FP], 4                          #85
    tv127-(EAX) = mov t126(EAX)                                  #85
    tv127-(EAX)<*23>, EFLAGS = add tv127-(EAX), &_amp;c           #85
    [tv127-(EAX)] = mov 1                                        #85
                jmp $L14                                         #85
$L14: (2 refs)                                                    #85
    tv129-(EAX) = mov 1                                          #86
    tv129-(EAX)<*24>, EFLAGS = add tv129-(EAX), _i[_FP]         #86
    _i[_FP] = mov tv129-(EAX)                                    #86
                jmp $L6                                          #87
$L7: (1 ref)                                                      #81
    [ESP], {ESP} = push &_amp;a, {ESP}                              #89
    [ESP], {ESP} = push &_amp;b, {ESP}                              #89
    t140(EAX) = lea &_amp;q[_FP]                                    #89
    [ESP], {ESP} = push t140(EAX), {ESP}                         #89
    [ESP], {ESP} = push 1, {ESP}                                  #89
    {-5}, {EAX ECX ESP EFLAGS} = call _try, $out[ESP], $out[ESP]+32, $out[ESP]
]+64, $out[ESP]+96, {-5}, {EAX ECX ESP EFLAGS} [Handler: $L1]
#89
                jmp $L27                                         #109
$L27: (1 ref)                                                      #109
    ESP, EFLAGS = add ESP, 16(0x00000010)                        #89
                jmp $L23                                         #109
$L23: (1 ref)                                                      #109
    t131(EFLAGS) = cmp(EQ) _q[_FP], 1                            #90
                jcc(EQ) t131(EFLAGS), $L17, $L16                #90
$L17: (1 ref)                                                      #90
    _i[_FP] = mov 1                                              #91
                jmp $L18                                         #92
$L18: (2 refs)                                                    #92
    t132(EFLAGS) = cmp(LE) _i[_FP], 8                            #92
                jcc(LE) t132(EFLAGS), $L20, $L19                #92
$L20: (1 ref)                                                      #92
    tv133-(EAX) = mov 1                                          #93
    tv133-(EAX)<*27>, EFLAGS = add tv133-(EAX), _i[_FP]         #93
    _i[_FP] = mov tv133-(EAX)                                    #93
                jmp $L18                                         #93
$L19: (1 ref)                                                      #92
                jmp $L21                                         #95
$L16: (1 ref)                                                      #90
    [ESP], {ESP} = push &$SG1197, {ESP}                         #97

```

AppendixF.txt

```

    {-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7 FPUStatus} = call
    _printf, $out[ESP], {-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7
    FPUStatus} [Handler: $L1]
    #97
        jmp $L28 #109
$L28: (1 ref) #109
    ESP, EFLAGS = add ESP, 4 #97
        jmp $L24 #109
$L24: (1 ref) #109
    [ESP], {ESP} = push 0, {ESP} #98
    {-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7 FPUStatus} = call
    _exit, $out[ESP], {-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7 F
    PUStatus} [Handler: $L1]
    #98
        jmp $L29 #109
$L29: (1 ref) #109
    ESP, EFLAGS = add ESP, 4 #98
        jmp $L25 #109
$L25: (1 ref) #109
        jmp $L21 #95
$L21: (2 refs) #95
        jmp $L3 #100
$L4: (1 ref) #78
    [ESP], {ESP} = push &$SG1198, {ESP} #101
    {-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7 FPUStatus} = call
    _printf, $out[ESP], {-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7
    FPUStatus} [Handler: $L1]
    #101
        jmp $L30 #109
$L30: (1 ref) #109
    ESP, EFLAGS = add ESP, 4 #101
        jmp $L26 #109
$L26: (1 ref) #109
    t142(EAX) = mov 0 #108
        jmp $L22 #108
$L1: (4 refs) #68
    UNWIND _main #68
$L22: (1 ref) #108
    EPILOGSTART #109
    EXITFUNC _main, t142(EAX) #109
    END _main, {-7} #109

```

IR after Frame Generation (flag Frame)

```

    {-7} = START _main #68
        ENTERFUNC _main #68
    [ESP], {ESP} = push EBP, {ESP} #68
    EBP = mov ESP #68
    ESP, EFLAGS = sub ESP, 16(0x00000010) #68
        PROLOGEND #68
    _doitagain[EBP] = mov 1 #78
        jmp $L2 #78

```


AppendixF.txt

```

$L3:  (1 ref)                                     #78
      t109(EAX) = mov _doitagain[EBP]             #78
      tv110-(EAX) = mov t109(EAX)                 #78
      tv110-(EAX)<*5>, EFLAGS = add tv110-(EAX), 1  #78
      _doitagain[EBP] = mov tv110-(EAX)           #78
      jmp $L2                                     #78
$L2:  (2 refs)                                     #78
      t111(EFLAGS) = cmp(LE) _doitagain[EBP], 6000(0x00001770) #78
      jcc(LE) t111(EFLAGS), $L5, $L4              #78
$L5:  (1 ref)                                     #78
      _i[EBP] = mov 0                             #80
      jmp $L6                                     #81
$L6:  (2 refs)                                     #81
      t112(EFLAGS) = cmp(LE) _i[EBP], 16(0x00000010) #81
      jcc(LE) t112(EFLAGS), $L8, $L7              #81
$L8:  (1 ref)                                     #81
      t113(EFLAGS) = cmp(GE) _i[EBP], 1           #83
      jcc(GE) t113(EFLAGS), $L10, $L9             #83
$L10: (1 ref)                                     #83
      t114(EFLAGS) = cmp(LE) _i[EBP], 8           #83
      jcc(LE) t114(EFLAGS), $L11, $L9             #83
$L11: (1 ref)                                     #83
      t116(EAX), EFLAGS = imul _i[EBP], 4         #83
      tv117-(EAX) = mov t116(EAX)                 #83
      tv117-(EAX)<*17>, EFLAGS = add tv117-(EAX), &_amp;a #83
      [tv117-(EAX)] = mov 1                       #83
      jmp $L9                                     #83
$L9:  (3 refs)                                     #83
      t119(EFLAGS) = cmp(GE) _i[EBP], 2           #84
      jcc(GE) t119(EFLAGS), $L13, $L12            #84
$L13: (1 ref)                                     #84
      t121(EAX), EFLAGS = imul _i[EBP], 4         #84
      tv122-(EAX) = mov t121(EAX)                 #84
      tv122-(EAX)<*20>, EFLAGS = add tv122-(EAX), &_amp;b #84
      [tv122-(EAX)] = mov 1                       #84
      jmp $L12                                     #84
$L12: (2 refs)                                     #84
      t124(EFLAGS) = cmp(LE) _i[EBP], 14(0x0000000e) #85
      jcc(LE) t124(EFLAGS), $L15, $L14            #85
$L15: (1 ref)                                     #85
      t126(EAX), EFLAGS = imul _i[EBP], 4         #85
      tv127-(EAX) = mov t126(EAX)                 #85
      tv127-(EAX)<*23>, EFLAGS = add tv127-(EAX), &_amp;c #85
      [tv127-(EAX)] = mov 1                       #85
      jmp $L14                                     #85
$L14: (2 refs)                                     #85
      tv129-(EAX) = mov 1                         #86
      tv129-(EAX)<*24>, EFLAGS = add tv129-(EAX), _i[EBP] #86
      _i[EBP] = mov tv129-(EAX)                   #86
      jmp $L6                                     #87
$L7:  (1 ref)                                     #81
      [ESP], {ESP} = push &_amp;a, {ESP}             #89
      [ESP], {ESP} = push &_amp;b, {ESP}             #89

```

AppendixF.txt

```

t140(EAX)    = lea &_q[EBP]                                #89
[ESP], {ESP} = push t140(EAX), {ESP}                      #89
[ESP], {ESP} = push 1, {ESP}                              #89
{-5}, {EAX ECX ESP EFLAGS} = call _try, $out[ESP], $out[ESP]+32, $out[ESP
]+64, $out[ESP]+96, {-5}, {EAX ECX ESP EFLAGS} [Handler: $L1]
#89
        jmp $L27                                           #109
$L27: (1 ref)                                           #109
    ESP, EFLAGS = add ESP, 16(0x00000010)                #89
        jmp $L23                                           #109
$L23: (1 ref)                                           #109
    t131(EFLAGS) = cmp(EQ) _q[EBP], 1                    #90
        jcc(EQ) t131(EFLAGS), $L17, $L16                 #90
$L17: (1 ref)                                           #90
    _i[EBP]      = mov 1                                  #91
        jmp $L18                                           #92
$L18: (2 refs)                                           #92
    t132(EFLAGS) = cmp(LE) _i[EBP], 8                    #92
        jcc(LE) t132(EFLAGS), $L20, $L19                 #92
$L20: (1 ref)                                           #92
    tv133-(EAX) = mov 1                                  #93
    tv133-(EAX)<*27>, EFLAGS = add tv133-(EAX), _i[EBP]  #93
    _i[EBP]      = mov tv133-(EAX)                      #93
        jmp $L18                                           #93
$L19: (1 ref)                                           #92
        jmp $L21                                           #95
$L16: (1 ref)                                           #90
    [ESP], {ESP} = push &$SG1197, {ESP}                  #97
    {-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7 FPUStatus} = call
_printf, $out[ESP], {-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7
FPUStatus} [Handler: $L1]
    #97
        jmp $L28                                           #109
$L28: (1 ref)                                           #109
    ESP, EFLAGS = add ESP, 4                             #97
        jmp $L24                                           #109
$L24: (1 ref)                                           #109
    [ESP], {ESP} = push 0, {ESP}                          #98
    {-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7 FPUStatus} = call
_exit, $out[ESP], {-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7 F
PUStatus} [Handler: $L1]
    #98
        jmp $L29                                           #109
$L29: (1 ref)                                           #109
    ESP, EFLAGS = add ESP, 4                             #98
        jmp $L25                                           #109
$L25: (1 ref)                                           #109
        jmp $L21                                           #95
$L21: (2 refs)                                           #95
        jmp $L3                                             #100
$L4: (1 ref)                                           #78
    [ESP], {ESP} = push &$SG1198, {ESP}                  #101
    {-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7 FPUStatus} = call

```

AppendixF.txt

```

_printf, $out[ESP], {-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7
_FPUStatus} [Handler: $L1]
#101
        jmp $L30
$L30:   (1 ref)
        ESP, EFLAGS = add ESP, 4
        jmp $L26
$L26:   (1 ref)
        t142(EAX) = mov 0
        jmp $L22
$L1:    (4 refs)
        UNWIND _main
$L22:   (1 ref)
        EPILOGSTART
        ESP = mov EBP
        EBP, {ESP} = pop [ESP], {ESP}
        {ESP} = ret {ESP}
        EXITFUNC _main, t142(EAX)
        END _main, {-7}

```

IR after Switch Lower (flag SwitchLower)

```

{-7} = START _main
        ENTERFUNC _main
[ESP], {ESP} = push EBP, {ESP}
EBP = mov ESP
ESP, EFLAGS = sub ESP, 16(0x00000010)
        PROLOGEND
_doitagain[EBP] = mov 1
        jmp $L2
$L3:    (1 ref)
        t109(EAX) = mov _doitagain[EBP]
        tv110-(EAX) = mov t109(EAX)
        tv110-(EAX)<*5>, EFLAGS = add tv110-(EAX), 1
        _doitagain[EBP] = mov tv110-(EAX)
        jmp $L2
$L2:    (2 refs)
        t111(EFLAGS) = cmp(LE) _doitagain[EBP], 6000(0x00001770)
        jcc(LE) t111(EFLAGS), $L5, $L4
$L5:    (1 ref)
        _i[EBP] = mov 0
        jmp $L6
$L6:    (2 refs)
        t112(EFLAGS) = cmp(LE) _i[EBP], 16(0x00000010)
        jcc(LE) t112(EFLAGS), $L8, $L7
$L8:    (1 ref)
        t113(EFLAGS) = cmp(GE) _i[EBP], 1
        jcc(GE) t113(EFLAGS), $L10, $L9
$L10:   (1 ref)
        t114(EFLAGS) = cmp(LE) _i[EBP], 8
        jcc(LE) t114(EFLAGS), $L11, $L9
$L11:   (1 ref)

```

AppendixF.txt

```

t116(EAX), EFLAGS = imul _i[EBP], 4 #83
tv117-(EAX) = mov t116(EAX) #83
tv117-(EAX)<*17>, EFLAGS = add tv117-(EAX), &_a #83
[tv117-(EAX)] = mov 1 #83
jmp $L9 #83
$L9: (3 refs) #83
t119(EFLAGS) = cmp(GE) _i[EBP], 2 #84
jcc(GE) t119(EFLAGS), $L13, $L12 #84
$L13: (1 ref) #84
t121(EAX), EFLAGS = imul _i[EBP], 4 #84
tv122-(EAX) = mov t121(EAX) #84
tv122-(EAX)<*20>, EFLAGS = add tv122-(EAX), &_b #84
[tv122-(EAX)] = mov 1 #84
jmp $L12 #84
$L12: (2 refs) #84
t124(EFLAGS) = cmp(LE) _i[EBP], 14(0x0000000e) #85
jcc(LE) t124(EFLAGS), $L15, $L14 #85
$L15: (1 ref) #85
t126(EAX), EFLAGS = imul _i[EBP], 4 #85
tv127-(EAX) = mov t126(EAX) #85
tv127-(EAX)<*23>, EFLAGS = add tv127-(EAX), &_c #85
[tv127-(EAX)] = mov 1 #85
jmp $L14 #85
$L14: (2 refs) #85
tv129-(EAX) = mov 1 #86
tv129-(EAX)<*24>, EFLAGS = add tv129-(EAX), _i[EBP] #86
_i[EBP] = mov tv129-(EAX) #86
jmp $L6 #87
$L7: (1 ref) #81
[ESP], {ESP} = push &_a, {ESP} #89
[ESP], {ESP} = push &_b, {ESP} #89
t140(EAX) = lea _q[EBP] #89
[ESP], {ESP} = push t140(EAX), {ESP} #89
[ESP], {ESP} = push 1, {ESP} #89
{-5}, {EAX ECX ESP EFLAGS} = call _try, $out[ESP], $out[ESP]+32, $out[ESP]
]+64, $out[ESP]+96, {-5}, {EAX ECX ESP EFLAGS} [Handler: $L1]
#89
jmp $L27 #109
$L27: (1 ref) #109
ESP, EFLAGS = add ESP, 16(0x00000010) #89
jmp $L23 #109
$L23: (1 ref) #109
t131(EFLAGS) = cmp(EQ) _q[EBP], 1 #90
jcc(EQ) t131(EFLAGS), $L17, $L16 #90
$L17: (1 ref) #90
_i[EBP] = mov 1 #91
jmp $L18 #92
$L18: (2 refs) #92
t132(EFLAGS) = cmp(LE) _i[EBP], 8 #92
jcc(LE) t132(EFLAGS), $L20, $L19 #92
$L20: (1 ref) #92
tv133-(EAX) = mov 1 #93
tv133-(EAX)<*27>, EFLAGS = add tv133-(EAX), _i[EBP] #93

```

```

    _i[EBP]      = mov tv133-(EAX)      #93
                    jmp $L18            #93
$L19: (1 ref)    jmp $L21              #92
                    jmp $L21            #95
$L16: (1 ref)    #90
    [ESP], {ESP} = push &$SG1197, {ESP} #97
    {-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7 FPUStatus} = call
    _printf, $out[ESP], {-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7
    FPUStatus} [Handler: $L1]
    #97
                    jmp $L28            #109
$L28: (1 ref)    #109
    ESP, EFLAGS = add ESP, 4           #97
                    jmp $L24            #109
$L24: (1 ref)    #109
    [ESP], {ESP} = push 0, {ESP}       #98
    {-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7 FPUStatus} = call
    _exit, $out[ESP], {-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7 F
    PUStatus} [Handler: $L1]
    #98
                    jmp $L29            #109
$L29: (1 ref)    #109
    ESP, EFLAGS = add ESP, 4           #98
                    jmp $L25            #109
$L25: (1 ref)    #109
                    jmp $L21            #95
$L21: (2 refs)   #95
                    jmp $L3            #100
$L4: (1 ref)     #78
    [ESP], {ESP} = push &$SG1198, {ESP} #101
    {-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7 FPUStatus} = call
    _printf, $out[ESP], {-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7
    FPUStatus} [Handler: $L1]
    #101
                    jmp $L30            #109
$L30: (1 ref)    #109
    ESP, EFLAGS = add ESP, 4           #101
                    jmp $L26            #109
$L26: (1 ref)    #109
    t142(EAX)     = mov 0              #108
                    jmp $L22            #108
$L1: (4 refs)    #68
    UNWIND _main    #68
$L22: (1 ref)    #108
    EPILOGSTART    #109
    ESP            = mov EBP          #109
    EBP, {ESP}     = pop [ESP], {ESP}  #109
    {ESP}          = ret {ESP}        #109
    EXITFUNC _main, t142(EAX)         #109
    END _main, {-7}                  #109

```

AppendixF.txt

```

{-7}          = START _main                                #68
                ENTERFUNC _main                            #68
[ESP], {ESP} = push EBP, {ESP}                             #68
EBP           = mov ESP                                    #68
ESP, EFLAGS = sub ESP, 16(0x00000010)                      #68
                PROLOGEND                                  #68
_doitagain[EBP] = mov 1                                    #78
                jmp $L2                                    #78
$L2: (2 refs)                                           #78
    t111(EFLAGS) = cmp(LE) _doitagain[EBP], 6000(0x00001770) #78
                jcc(LE) t111(EFLAGS), $L5                 #78
                jmp $L4                                    #78
$L5: (1 ref)                                           #78
    _i[EBP]     = mov 0                                    #80
                jmp $L6                                    #81
$L6: (2 refs)                                           #81
    t112(EFLAGS) = cmp(LE) _i[EBP], 16(0x00000010)         #81
                jcc(LE) t112(EFLAGS), $L8                 #81
                jmp $L7                                    #81
$L8: (1 ref)                                           #81
    t113(EFLAGS) = cmp(GE) _i[EBP], 1                     #83
                jcc(GE) t113(EFLAGS), $L10                #83
                jmp $L9                                    #83
$L10: (1 ref)                                           #83
    t114(EFLAGS) = cmp(LE) _i[EBP], 8                     #83
                jcc(LE) t114(EFLAGS), $L11                #83
                jmp $L9                                    #83
$L11: (1 ref)                                           #83
    t116(EAX), EFLAGS = imul _i[EBP], 4                   #83
    tv117-(EAX) = mov t116(EAX)                           #83
    tv117-(EAX)<*17>, EFLAGS = add tv117-(EAX), &_a         #83
    [tv117-(EAX)] = mov 1                                  #83
                jmp $L9                                    #83
$L9: (3 refs)                                           #83
    t119(EFLAGS) = cmp(GE) _i[EBP], 2                     #84
                jcc(GE) t119(EFLAGS), $L13                #84
                jmp $L12                                    #84
$L13: (1 ref)                                           #84
    t121(EAX), EFLAGS = imul _i[EBP], 4                   #84
    tv122-(EAX) = mov t121(EAX)                           #84
    tv122-(EAX)<*20>, EFLAGS = add tv122-(EAX), &_b         #84
    [tv122-(EAX)] = mov 1                                  #84
                jmp $L12                                    #84
$L12: (2 refs)                                           #84
    t124(EFLAGS) = cmp(LE) _i[EBP], 14(0x0000000e).       #85
                jcc(LE) t124(EFLAGS), $L15                #85
                jmp $L14                                    #85
$L15: (1 ref)                                           #85
    t126(EAX), EFLAGS = imul _i[EBP], 4                   #85
    tv127-(EAX) = mov t126(EAX)                           #85
    tv127-(EAX)<*23>, EFLAGS = add tv127-(EAX), &_c         #85
    [tv127-(EAX)] = mov 1                                  #85

```

```

                                jmp $L14                                #85
$L14: (2 refs)                                #85
    tv129-(EAX) = mov 1                                #86
    tv129-(EAX)<*24>, EFLAGS = add tv129-(EAX), _i[EBP]    #86
    _i[EBP] = mov tv129-(EAX)                        #86
                                jmp $L6                                #87
$L7: (1 ref)                                #81
    [ESP], {ESP} = push &_a, {ESP}                    #89
    [ESP], {ESP} = push &_b, {ESP}                    #89
    t140(EAX) = lea &_q[EBP]                          #89
    [ESP], {ESP} = push t140(EAX), {ESP}                #89
    [ESP], {ESP} = push 1, {ESP}                      #89
    {-5}, {EAX ECX ESP EFLAGS} = call _try, $out[ESP], $out[ESP]+32, $out[ESP
]+64, $out[ESP]+96, {-5}, {EAX ECX ESP EFLAGS} [Handler: $L1]
#89
                                jmp $L27                                #109
$L27: (1 ref)                                #109
    ESP, EFLAGS = add ESP, 16(0x00000010)              #89
                                jmp $L23                                #109
$L23: (1 ref)                                #109
    t131(EFLAGS) = cmp(EQ) _q[EBP], 1                  #90
                                jcc(EQ) t131(EFLAGS), $L17          #90
                                jmp $L16                                #90
$L17: (1 ref)                                #90
    _i[EBP] = mov 1                                    #91
                                jmp $L18                                #92
$L18: (2 refs)                                #92
    t132(EFLAGS) = cmp(LE) _i[EBP], 8                  #92
                                jcc(LE) t132(EFLAGS), $L20          #92
                                jmp $L19                                #92
$L20: (1 ref)                                #92
    tv133-(EAX) = mov 1                                #93
    tv133-(EAX)<*27>, EFLAGS = add tv133-(EAX), _i[EBP]    #93
    _i[EBP] = mov tv133-(EAX)                        #93
                                jmp $L18                                #93
$L19: (1 ref)                                #92
                                jmp $L21                                #95
$L16: (1 ref)                                #90
    [ESP], {ESP} = push &$SG1197, {ESP}                #97
    {-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7 FPUStatus} = call
_printf, $out[ESP], {-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7
FPUStatus} [Handler: $L1]
#97
                                jmp $L28                                #109
$L28: (1 ref)                                #109
    ESP, EFLAGS = add ESP, 4                            #97
                                jmp $L24                                #109
$L24: (1 ref)                                #109
    [ESP], {ESP} = push 0, {ESP}                      #98
    {-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7 FPUStatus} = call
_exit, $out[ESP], {-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7 F
PUStatus} [Handler: $L1]
#98

```

AppendixF.txt

```

                                jmp $L29                                #109
$L29: (1 ref)                                #109
    ESP, EFLAGS = add ESP, 4                                #98
                                jmp $L25                                #109
$L25: (1 ref)                                #109
                                jmp $L21                                #95
$L21: (2 refs)                                #95
                                jmp $L3                                #100
$L3: (1 ref)                                #78
    t109(EAX) = mov _doitagain[EBP]                                #78
    tv110-(EAX) = mov t109(EAX)                                #78
    tv110-(EAX)<*5>, EFLAGS = add tv110-(EAX), 1                #78
    _doitagain[EBP] = mov tv110-(EAX)                            #78
                                jmp $L2                                #78
$L4: (1 ref)                                #78
    [ESP], {ESP} = push &$SG1198, {ESP}                        #101
    {-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7 FPUS
_status} = call
    _printf, $out[ESP], {-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7
    FPUS_status} [Handler: $L1]
    #101
                                jmp $L30                                #109
$L30: (1 ref)                                #109
    ESP, EFLAGS = add ESP, 4                                #101
                                jmp $L26                                #109
$L26: (1 ref)                                #109
    t142(EAX) = mov 0                                #108
                                jmp $L22                                #108
$L22: (1 ref)                                #108
    EPILOGSTART                                #109
    ESP = mov EBP                                #109
    EBP, {ESP} = pop [ESP], {ESP}                            #109
    {ESP} = ret {ESP}                                #109
    EXITFUNC _main, t142(EAX)                                #109
$L1: (4 refs)                                #68
    UNWIND _main                                #68
    END _main, {-7}                                #109

```

IR after Flow Optimization (flag FlowOpts)

```

    {-7} = START _main                                #68
    ENTERFUNC _main                                #68
    [ESP], {ESP} = push EBP, {ESP}                    #68
    EBP = mov ESP                                #68
    ESP, EFLAGS = sub ESP, 16(0x00000010)                #68
    PROLOGEND                                #68
    _doitagain[EBP] = mov 1                                #78
$L2: (1 ref)                                #78
    t111(EFLAGS) = cmp(LE) _doitagain[EBP], 6000(0x00001770)    #78
    jcc(GT) t111(EFLAGS), $L4                                #78
    _i[EBP] = mov 0                                #80
$L6: (1 ref)                                #81
    t112(EFLAGS) = cmp(LE) _i[EBP], 16(0x00000010)            #81

```


AppendixF.txt

```

                                jcc(GT) t112(EFLAGS), $L7                #81
t113(EFLAGS) = cmp(GE) _i[EBP], 1                                     #83
                                jcc(LT) t113(EFLAGS), $L9                #83
t114(EFLAGS) = cmp(LE) _i[EBP], 8                                     #83
                                jcc(GT) t114(EFLAGS), $L9                #83
t116(EAX), EFLAGS = imul _i[EBP], 4                                   #83
tv117-(EAX) = mov t116(EAX)                                           #83
tv117-(EAX)<*17>, EFLAGS = add tv117-(EAX), &_amp;a                     #83
[tv117-(EAX)] = mov 1                                                 #83
$L9: (2 refs)                                                         #83
    t119(EFLAGS) = cmp(GE) _i[EBP], 2                                   #84
                                jcc(LT) t119(EFLAGS), $L12              #84
    t121(EAX), EFLAGS = imul _i[EBP], 4                                   #84
    tv122-(EAX) = mov t121(EAX)                                         #84
    tv122-(EAX)<*20>, EFLAGS = add tv122-(EAX), &_amp;b                     #84
    [tv122-(EAX)] = mov 1                                              #84
$L12: (1 ref)                                                         #84
    t124(EFLAGS) = cmp(LE) _i[EBP], 14(0x0000000e)                   #85
                                jcc(GT) t124(EFLAGS), $L14              #85
    t126(EAX), EFLAGS = imul _i[EBP], 4                                   #85
    tv127-(EAX) = mov t126(EAX)                                         #85
    tv127-(EAX)<*23>, EFLAGS = add tv127-(EAX), &_amp;c                     #85
    [tv127-(EAX)] = mov 1                                              #85
$L14: (1 ref)                                                         #85
    tv129-(EAX) = mov 1                                                 #86
    tv129-(EAX)<*24>, EFLAGS = add tv129-(EAX), _i[EBP]               #86
    _i[EBP] = mov tv129-(EAX)                                           #86
                                jmp $L6                                  #87
$L7: (1 ref)                                                         #81
    [ESP], {ESP} = push &_amp;a, {ESP}                                     #89
    [ESP], {ESP} = push &_amp;b, {ESP}                                     #89
    t140(EAX) = lea &_amp;q[EBP]                                           #89
    [ESP], {ESP} = push t140(EAX), {ESP}                                #89
    [ESP], {ESP} = push 1, {ESP}                                         #89
    {-5}, {EAX ECX ESP EFLAGS} = call _try, $out[ESP], $out[ESP]+32, $out[ESP]
]+64, $out[ESP]+96, {-5}, {EAX ECX ESP EFLAGS} [Handler: $L1]
#89
    ESP, EFLAGS = add ESP, 16(0x00000010)                               #89
    t131(EFLAGS) = cmp(EQ) _q[EBP], 1                                   #90
                                jcc(NE) t131(EFLAGS), $L16              #90
    _i[EBP] = mov 1                                                     #91
$L18: (1 ref)                                                         #92
    t132(EFLAGS) = cmp(LE) _i[EBP], 8                                   #92
                                jcc(GT) t132(EFLAGS), $L3               #92
    tv133-(EAX) = mov 1                                                 #93
    tv133-(EAX)<*27>, EFLAGS = add tv133-(EAX), _i[EBP]               #93
    _i[EBP] = mov tv133-(EAX)                                           #93
                                jmp $L18                                  #93
$L16: (1 ref)                                                         #90
    [ESP], {ESP} = push &$SG1197, {ESP}                                #97
    {-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7 FPUS
tatus} = call _printf, $out[ESP], {-5}, {EAX ECX EDX ESP EFLA
GS MM0-MM7 XMM0-XMM7 FP0-FP7 FPUS
tatus} [Handler: $L1]

```

AppendixF.txt

```

#97
ESP, EFLAGS = add ESP, 4 #97
[ESP], {ESP} = push 0, {ESP} #98
{-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7 FPUStatus} = call
exit, $out[ESP], {-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7 F
PUSStatus} [Handler: $L1]
#98
ESP, EFLAGS = add ESP, 4 #98
$L3: (1 ref) #78
    t109(EAX) = mov _doitagain[EBP] #78
    tv110-(EAX) = mov t109(EAX) #78
    tv110-(EAX)<*5>, EFLAGS = add tv110-(EAX), 1 #78
    _doitagain[EBP] = mov tv110-(EAX) #78
    jmp $L2 #78
$L4: (1 ref) #78
    [ESP], {ESP} = push &$SG1198, {ESP} #101
    {-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7 FPUStatus} = call
_printf, $out[ESP], {-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7
FPUStatus} [Handler: $L1]
#101
ESP, EFLAGS = add ESP, 4 #101
t142(EAX) = mov 0 #108
    EPILOGSTART #109
ESP = mov EBP #109
EBP, {ESP} = pop [ESP], {ESP} #109
{ESP} = ret {ESP} #109
    EXITFUNC _main, t142(EAX) #109
$L1: (4 refs) #68
    UNWIND _main #68
    END _main, {-7} #109

```

IR after Encoding, Listing, COFF Emission (flag Encode)

```

{-7} = START _main #68
    ENTERFUNC _main #68
[ESP], {ESP} = push EBP, {ESP} #68
EBP = mov ESP #68
ESP, EFLAGS = sub ESP, 16(0x00000010) #68
    PROLOGEND #68
    _doitagain[EBP] = mov 1 #78
$L2: (1 ref) #78
    t111(EFLAGS) = cmp(LE) _doitagain[EBP], 6000(0x00001770) #78
    jg(GT) t111(EFLAGS), $L4 #78
    _i[EBP] = mov 0 #80
$L6: (1 ref) #81
    t112(EFLAGS) = cmp(LE) _i[EBP], 16(0x00000010) #81
    jg(GT) t112(EFLAGS), $L7 #81
    t113(EFLAGS) = cmp(GE) _i[EBP], 1 #83
    jl(LT) t113(EFLAGS), $L9 #83
    t114(EFLAGS) = cmp(LE) _i[EBP], 8 #83
    jg(GT) t114(EFLAGS), $L9 #83
    tv116-(EAX)<*16>, EFLAGS = imul _i[EBP], 4 #83

```

AppendixF.txt

```

tv117-(EAX)<*17>, EFLAGS = add tv117-(EAX), &_a      #83
[tv117-(EAX)] = mov 1                                #83
$L9:  (2 refs)                                       #83
    t119(EFLAGS) = cmp(GE) _i[EBP], 2                #84
                    jl(LT) t119(EFLAGS), $L12         #84
    tv121-(EAX)<*19>, EFLAGS = imul _i[EBP], 4         #84
    tv122-(EAX)<*20>, EFLAGS = add tv122-(EAX), &_b   #84
    [tv122-(EAX)] = mov 1                             #84
$L12:  (1 ref)                                       #84
    t124(EFLAGS) = cmp(LE) _i[EBP], 14(0x0000000e)   #85
                    jg(GT) t124(EFLAGS), $L14         #85
    tv126-(EAX)<*22>, EFLAGS = imul _i[EBP], 4         #85
    tv127-(EAX)<*23>, EFLAGS = add tv127-(EAX), &_c   #85
    [tv127-(EAX)] = mov 1                             #85
$L14:  (1 ref)                                       #85
    tv129-(EAX) = mov 1                               #86
    tv129-(EAX)<*24>, EFLAGS = add tv129-(EAX), _i[EBP] #86
    _i[EBP]      = mov tv129-(EAX)                   #86
                    jmp $L6                             #87
$L7:  (1 ref)                                       #81
    [ESP], {ESP} = push &_a, {ESP}                   #89
    [ESP], {ESP} = push &_b, {ESP}                   #89
    t140(EAX)     = lea &_q[EBP]                      #89
    [ESP], {ESP} = push t140(EAX), {ESP}              #89
    [ESP], {ESP} = push 1, {ESP}                     #89
    {-5}, {EAX ECX ESP EFLAGS} = call _try, $out[ESP], $out[ESP]+32, $out[ESP]
] +64, $out[ESP]+96, {-5}, {EAX ECX ESP EFLAGS} [Handler: $L1]
#89
    ESP, EFLAGS = add ESP, 16(0x00000010)             #89
    t131(EFLAGS) = cmp(EQ) _q[EBP], 1                 #90
                    jne(NE) t131(EFLAGS), $L16         #90
    _i[EBP]      = mov 1                             #91
$L18:  (1 ref)                                       #92
    t132(EFLAGS) = cmp(LE) _i[EBP], 8                 #92
                    jg(GT) t132(EFLAGS), $L3           #92
    tv133-(EAX) = mov 1                               #93
    tv133-(EAX)<*27>, EFLAGS = add tv133-(EAX), _i[EBP] #93
    _i[EBP]      = mov tv133-(EAX)                   #93
                    jmp $L18                           #93
$L16:  (1 ref)                                       #90
    [ESP], {ESP} = push &$SG1197, {ESP}               #97
    {-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7 FPUStatus} = call
_printf, $out[ESP], {-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7
_FPUStatus} [Handler: $L1]
#97
    ESP, EFLAGS = add ESP, 4                          #97
    [ESP], {ESP} = push 0, {ESP}                      #98
    {-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7 FPUStatus} = call
_exit, $out[ESP], {-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7 F
PUStatus} [Handler: $L1]
#98
    ESP, EFLAGS = add ESP, 4                          #98
$L3:  (1 ref)                                       #78

```

AppendixF.txt

```

tv109-(EAX)<*4> = mov _doitagain[EBP] #78
tv110-(EAX)<*5>, EFLAGS = add tv110-(EAX), 1 #78
_doitagain[EBP] = mov tv110-(EAX) #78
        jmp $L2 #78
$L4: (1 ref) #78
    [ESP], {ESP} = push &$SG1198, {ESP} #101
    {-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7 FPUStatus} = call
    _printf, $out[ESP], {-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7
    FPUStatus} [Handler: $L1]
    #101
    ESP, EFLAGS = add ESP, 4 #101
    t142(EAX) = mov 0 #108
        EPILOGSTART #109
    ESP = mov EBP #109
    EBP, {ESP} = pop [ESP], {ESP} #109
    {ESP} = ret {ESP} #109
        EXITFUNC _main, t142(EAX) #109
$L1: (4 refs) #68
        UNWIND _main #68
        END _main, {-7} #109

```

IR after LIR Phases (flag LIR Phases) [SubPhaseList]

```

{-7} = START _main #68
        ENTERFUNC _main #68
[ESP], {ESP} = push EBP, {ESP} #68
EBP = mov ESP #68
ESP, EFLAGS = sub ESP, 16(0x00000010) #68
        PROLOGEND #68
    _doitagain[EBP] = mov 1 #78
$L2: (1 ref) #78
    t111(EFLAGS) = cmp(LE) _doitagain[EBP], 6000(0x00001770) #78
        jg(GT) t111(EFLAGS), $L4 #78
    _i[EBP] = mov 0 #80
$L6: (1 ref) #81
    t112(EFLAGS) = cmp(LE) _i[EBP], 16(0x00000010) #81
        jg(GT) t112(EFLAGS), $L7 #81
    t113(EFLAGS) = cmp(GE) _i[EBP], 1 #83
        jl(LT) t113(EFLAGS), $L9 #83
    t114(EFLAGS) = cmp(LE) _i[EBP], 8 #83
        jg(GT) t114(EFLAGS), $L9 #83
    tv116-(EAX)<*16>, EFLAGS = imul _i[EBP], 4 #83
    tv117-(EAX)<*17>, EFLAGS = add tv117-(EAX), &_a #83
    [tv117-(EAX)] = mov 1 #83
$L9: (2 refs) #83
    t119(EFLAGS) = cmp(GE) _i[EBP], 2 #84
        jl(LT) t119(EFLAGS), $L12 #84
    tv121-(EAX)<*19>, EFLAGS = imul _i[EBP], 4 #84
    tv122-(EAX)<*20>, EFLAGS = add tv122-(EAX), &_b #84
    [tv122-(EAX)] = mov 1 #84
$L12: (1 ref) #84
    t124(EFLAGS) = cmp(LE) _i[EBP], 14(0x0000000e) #85

```

```

AppendixF.txt
    jg(GT) t124(EFLAGS), $L14                                #85
tv126-(EAX)<*22>, EFLAGS = imul _i[EBP], 4                    #85
tv127-(EAX)<*23>, EFLAGS = add tv127-(EAX), &_c                #85
[tv127-(EAX)] = mov 1                                        #85
$L14: (1 ref)                                                #85
    tv129-(EAX) = mov 1                                        #86
    tv129-(EAX)<*24>, EFLAGS = add tv129-(EAX), _i[EBP]      #86
    _i[EBP] = mov tv129-(EAX)                                #86
    jmp $L6                                                    #87
$L7: (1 ref)                                                  #81
    [ESP], {ESP} = push &_a, {ESP}                             #89
    [ESP], {ESP} = push &_b, {ESP}                             #89
    t140(EAX) = lea &_q[EBP]                                    #89
    [ESP], {ESP} = push t140(EAX), {ESP}                       #89
    [ESP], {ESP} = push 1, {ESP}                                #89
    {-5}, {EAX ECX ESP EFLAGS} = call _try, $out[ESP], $out[ESP]+32, $out[ESP
]+64, $out[ESP]+96, {-5}, {EAX ECX ESP EFLAGS} [Handler: $L1]
#89
    ESP, EFLAGS = add ESP, 16(0x00000010)                     #89
    t131(EFLAGS) = cmp(EQ) _q[EBP], 1                         #90
    jne(NE) t131(EFLAGS), $L16                                #90
    _i[EBP] = mov 1                                           #91
$L18: (1 ref)                                                  #92
    t132(EFLAGS) = cmp(LE) _i[EBP], 8                         #92
    jg(GT) t132(EFLAGS), $L3                                  #92
    tv133-(EAX) = mov 1                                        #93
    tv133-(EAX)<*27>, EFLAGS = add tv133-(EAX), _i[EBP]      #93
    _i[EBP] = mov tv133-(EAX)                                #93
    jmp $L18                                                    #93
$L16: (1 ref)                                                  #90
    [ESP], {ESP} = push &$SG1197, {ESP}                       #97
    {-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7 FPUStatus} = call
_printf, $out[ESP], {-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7
FPUStatus} [Handler: $L1]
#97
    ESP, EFLAGS = add ESP, 4                                   #97
    [ESP], {ESP} = push 0, {ESP}                               #98
    {-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7 FPUStatus} = call
_exit, $out[ESP], {-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7 F
PUStatus} [Handler: $L1]
#98
    ESP, EFLAGS = add ESP, 4                                   #98
$L3: (1 ref)                                                  #78
    tv109-(EAX)<*4> = mov _doitagain[EBP]                      #78
    tv110-(EAX)<*5>, EFLAGS = add tv110-(EAX), 1              #78
    _doitagain[EBP] = mov tv110-(EAX)                         #78
    jmp $L2                                                    #78
$L4: (1 ref)                                                  #78
    [ESP], {ESP} = push &$SG1198, {ESP}                       #101
    {-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7 FPUStatus} = call
_printf, $out[ESP], {-5}, {EAX ECX EDX ESP EFLAGS MM0-MM7 XMM0-XMM7 FP0-FP7
FPUStatus} [Handler: $L1]
#101

```

AppendixF.txt

ESP, EFLAGS	= add ESP, 4	#101
t142(EAX)	= mov 0	#108
	EPILOGSTART	#109
ESP	= mov EBP	#109
EBP, {ESP}	= pop [ESP], {ESP}	#109
{ESP}	= ret {ESP}	#109
	EXITFUNC _main, t142(EAX)	#109
\$L1: (4 refs)		#68
	UNWIND _main	#68
	END _main, {-7}	#109

B-61